Network Satisfaction Problems Solved by k-Consistency

Manuel Bodirsky
Institut für Algebra, TU Dresden, Germany

Simon Knäuer
Institut für Algebra, TU Dresden, Germany

Abstract

We show that the problem of deciding for a given finite relation algebra $\mathbf{A}$ whether the network satisfaction problem for $\mathbf{A}$ can be solved by the k-consistency procedure, for some $k \in \mathbb{N}$, is undecidable. For the important class of finite relation algebras $\mathbf{A}$ with a normal representation, however, the decidability of this problem remains open. We show that if $\mathbf{A}$ is symmetric and has a flexible atom, then the question whether NSP($\mathbf{A}$) can be solved by k-consistency, for some $k \in \mathbb{N}$, is decidable (even in polynomial time in the number of atoms of $\mathbf{A}$). This result follows from a more general sufficient condition for the correctness of the k-consistency procedure for finite symmetric relation algebras. In our proof we make use of a result of Alexandr Kazda about finite binary conservative structures.

2012 ACM Subject Classification Theory of computation → Finite Model Theory; Theory of computation → Problems, reductions and completeness; Theory of computation → Complexity theory and logic

Keywords and phrases Constraint Satisfaction, Computational Complexity, Relation Algebras, Network Satisfaction, Qualitative Reasoning, k-Consistency, Datalog

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.116

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version *Full Version*: `https://arxiv.org/abs/2304.12871` [13]

Funding *Manuel Bodirsky*: The author has received funding from the European Union (ERC Grant NO. 681988, CSP-Infinity and ERC Synergy Grant No. 101071674, POCOCOP).
Simon Knäuer: The author was supported by DFG Graduiertenkolleg 1763 (QuantLA).

1 Introduction

Many computational problems in qualitative temporal and spatial reasoning can be phrased as *network satisfaction problems (NSPs)* for *finite relation algebras*. Such a network consists of a finite set of nodes, and a labelling of pairs of nodes by elements of the relation algebra. In applications, such a network models some partial (and potentially inconsistent) knowledge that we have about some temporal or spatial configuration. The computational task is to replace the labels by *atoms* of the relation algebra such that the resulting network has an embedding into a representation of the relation algebra. In applications, this embedding provides a witness that the input configuration is consistent (a formal definition of relation algebras, representations, and the network satisfaction problem can be found in Section 2.1). The computational complexity of the network satisfaction problem depends on the fixed finite relation algebra, and is of central interest in the mentioned application areas. Relation algebras have been studied since the 40's with famous contributions of Tarski [41], Lyndon [34], McKenzie [37,38], and many others, with renewed interest since the 90s [7,11,22,25–27,30].

One of the most prominent algorithms for solving NSPs in polynomial time is the so-called *path consistency procedure*. The path consistency procedure has a natural generalisation to the *k-consistency procedure*, for some fixed $k \geq 3$. Such consistency algorithms have a number of advantages: e.g., they run in polynomial time, and they are *one-sided correct*, i.e., if they reject an instance, then we can be sure that the instance is unsatisfiable. Because of these properties, consistency algorithms can be used to prune the search space in exhaustive approaches that are used if the network consistency problem is NP-complete. The question for what temporal and spatial reasoning problems the k-consistency procedure provides a necessary and sufficient condition for satisfiability is among the most important research problems in the area [9, 40]. The analogous problem for so-called *constraint satisfaction problems (CSPs)* was posed by Feder and Vardi [23] and has been solved for finite-domain CSPs by Barto and Kozik [5]. Their result also shows that for a given finite-domain template, the question whether the corresponding CSP can be solved by the k-consistency procedure can be decided in polynomial time.

In contrast, we show that there is no algorithm that decides for a given finite relation algebra $\mathbf{A}$ whether NSP($\mathbf{A}$) can be solved by the k-consistency procedure, for some $k \in \mathbb{N}$. The question is also undecidable for every fixed $k \geq 3$; in particular, there is no algorithm that decides whether NSP($\mathfrak{A}$) can be solved by the path consistency procedure. Our proof relies on results of Hirsch [29] and Hirsch and Hodkinson [25]. The proof also shows that Hirsch's *Really Big Complexity Problem (RBCP; [27])* is undecidable. The RBCP asks for a description of those finite relation algebras $\mathbf{A}$ whose NSP can be solved in polynomial time.

Many of the classic examples of relation algebras that are used in temporal and spatial reasoning, such as the point algebra, Allen's Interval Algebra, RCC5, RCC8, have so-called *normal representations*, which are representations that are particularly well-behaved from a model theory perspective [7, 9, 27]. The importance of normal representations combined with our negative results for general finite relation algebras prompts the question whether solvability of the NSP by the k-consistency procedure can at least be characterised for relation algebras $\mathbf{A}$ with a normal representation. Our main result is a sufficient condition that implies that NSP($\mathbf{A}$) can be solved by the k-consistency procedure (Theorem 30). The condition can be checked algorithmically for a given $\mathbf{A}$. Moreover, for symmetric relation algebras with a flexible atom, which form a large subclass of the class of relation algebras with a normal representation, our condition provides a necessary and sufficient criterion for solvability by k-consistency (Theorem 39). We prove that the NSP for every symmetric relation algebra with a flexible atom that cannot be solved by the k-consistency procedure is already NP-complete. Finally, for symmetric relation algebras with a flexible atom our tractability condition can even be checked in polynomial time for a given relation algebra $\mathbf{A}$ (Theorem 42).

In our proof, we exploit a connection between the NSP for relation algebras $\mathbf{A}$ with a normal representation and finite-domain constraint satisfaction problems. In a next step, this allows us to use strong results for CSPs over finite domains. There are similarities between the fact that the set of relations of a representation of $\mathbf{A}$ is closed under taking unions on the one hand, and so-called *conservative finite-domain CSPs* [3, 16–18] on the other hand; in a conservative CSP the set of allowed constraints in instances of the CSP contains all unary relations. The complexity of conservative CSPs has been classified long before the solution of the Feder-Vardi Dichotomy Conjecture [19, 23, 42, 43]. Moreover, there are particularly elegant descriptions of when a finite-domain conservative CSP can be solved by the k-consistency procedure for some $k \in \mathbb{N}$ (see, e.g., Theorem 2.17 in [17]). Our approach is to turn the similarities into a formal correspondence so that we can use these results for finite-domain conservative CSPs to prove that k-consistency solves NSP($\mathbf{A}$). A key ingredient here is a contribution of Kazda [31] about conservative *binary* CSPs.

All the missing proofs and details can be found in an extended version of this article [13].

2 Preliminaries

A *signature* τ is a set of function or relation symbols each of which has an associated finite *arity* $k \in \mathbb{N}$. A τ-structure $\mathfrak{A}$ consists of a set A together with a function $f^{\mathfrak{A}} \colon A^k \to A$ for every function symbol $f \in \tau$ of arity k and a relation $R^{\mathfrak{A}} \subseteq A^k$ for every relation symbol $R \in \tau$ of arity k. The set A is called the *domain of* $\mathfrak{A}$. Let $\mathfrak{A}$ and $\mathfrak{B}$ be τ-structures. The *(direct) product* $\mathfrak{C} = \mathfrak{A} \times \mathfrak{B}$ is the τ-structure where

- $A \times B$ is the domain of $\mathfrak{C}$;
- for every relation symbol Q of arity $n \in \mathbb{N}$ and every tuple $((a_1, b_1), \dots, (a_n, b_n)) \in (A \times B)^n$, we have that $((a_1, b_1), \dots, (a_n, b_n)) \in Q^{\mathfrak{C}}$ if and only if $(a_1, \dots, a_n) \in Q^{\mathfrak{A}}$ and $(b_1, \dots, b_n) \in Q^{\mathfrak{B}}$;
- for every function symbol Q of arity $n \in \mathbb{N}$ and every tuple $((a_1, b_1), \dots, (a_n, b_n)) \in (A \times B)^n$, we have that

$$Q^{\mathfrak{C}}((a_1, b_1), \dots, (a_n, b_n)) := (Q^{\mathfrak{A}}(a_1, \dots, a_n), Q^{\mathfrak{B}}(b_1, \dots, b_n)).$$

We denote the (direct) product $\mathfrak{A} \times \mathfrak{A}$ by $\mathfrak{A}^2$. The k-fold product $\mathfrak{A} \times \cdots \times \mathfrak{A}$ is defined analogously and denoted by $\mathfrak{A}^k$. Structures with a signature that only contains function symbols are called *algebras* and structures with purely relational signature are called *relational structures*. Since we do not deal with signatures of mixed type in this article, we will use the term structure for relational structures only.

2.1 Relation Algebras

Relation algebras are particular algebras; in this section we recall their definition and state some of their basic properties. We introduce *proper* relation algebras, move on to abstract relation algebras, and finally define representations of relation algebras. For an introduction to relation algebras we recommend the textbook by Maddux [36]. Proper relation algebras are algebras whose domain is a set of binary relations over a common domain, and which are equipped with certain operations on binary relations.

▶ **Definition 1.** *Let D be a set and $\mathcal{R}$ a set of binary relations over D such that $(\mathcal{R}; \cup, ^{-}, 0, 1, \mathrm{Id}, ^{\smile}, \circ)$ is an algebra with operations defined as follows:*

1. $0 := \emptyset$,
2. $1 := \bigcup \mathcal{R}$,
3. $\mathrm{Id} := \{(x, x) \mid x \in D\}$,
4. $a \cup b := \{(x, y) \mid (x, y) \in a \vee (x, y) \in b\}$,
5. $\bar{a} := 1 \setminus a$,
6. $\breve{a} := \{(x, y) \mid (y, x) \in a\}$,
7. $a \circ b := \{(x, z) \mid \exists y \in D : (x, y) \in a \ \text{ and } \ (y, z) \in b\}$,

for $a, b \in \mathcal{R}$. Then $(\mathcal{R}; \cup, ^{-}, 0, 1, \mathrm{Id}, ^{\smile}, \circ)$ is called a proper relation algebra.

The class of all proper relation algebras is denoted by PA. Abstract relation algebras are a generalisation of proper relation algebras where the domain does not need to be a set of binary relations.

▶ **Definition 2.** *An* (abstract) relation algebra $\mathbf{A}$ *is an algebra with domain* A *and signature* $\{\cup, ^-, 0, 1, \mathrm{Id}, ^\smile, \circ\}$ *such that*

1. *the structure* $(A; \cup, \cap, ^-, 0, 1)$*, with* $\cap$ *defined by* $x \cap y := \overline{(\bar{x} \cup \bar{y})}$*, is a Boolean algebra,*
2. $\circ$ *is an associative binary operation on* A*, called* composition*,*
3. *for all* $a, b, c, \in A$*:* $(a \cup b) \circ c = (a \circ c) \cup (b \circ c)$*,*
4. *for all* $a \in A$*:* $a \circ \mathrm{Id} = a$*,*
5. *for all* $a \in A$*:* $\breve{\breve{a}} = a$*,*
6. *for all* $a, b \in A$*:* $\breve{x} = \breve{a} \cup \breve{b}$ *where* $x := a \cup b$*,*
7. *for all* $a, b \in A$*:* $\breve{x} = \breve{b} \circ \breve{a}$ *where* $x := a \circ b$*,*
8. *for all* $a, b, c \in A$*:* $\bar{b} \cup (\breve{a} \circ (\overline{(a \circ b)})) = \bar{b}$*.*

We denote the class of all relation algebras by RA. Let $\mathbf{A} = (A; \cup, ^-, 0, 1, \mathrm{Id}, ^\smile, \circ)$ be a relation algebra. By definition, $(A; \cup, \cap, ^-, 0, 1)$ is a Boolean algebra and therefore induces a partial order $\leq$ on A, which is defined by $x \leq y :\Leftrightarrow x \cup y = y$. Note that for proper relation algebras this ordering coincides with the set-inclusion order. The minimal elements of this order in $A \setminus \{0\}$ are called *atoms*. The set of atoms of $\mathbf{A}$ is denoted by A_0. Note that for the finite Boolean algebra $(A; \cup, \cap, ^-, 0, 1)$ each element $a \in A$ can be uniquely represented as the union $\cup$ (or "join") of elements from a subset of A_0. We will often use this fact and directly denote elements of the relation algebra $\mathbf{A}$ by subsets of A_0.

By item 3. in Definition 2 the values of the composition operation $\circ$ in $\mathbf{A}$ are completely determined by the values of $\circ$ on A_0. This means that for a finite relation algebra the operation $\circ$ can be represented by a multiplication table for the atoms A_0.

An algebra with signature $\tau = \{\cup, ^-, 0, 1, \mathrm{Id}, ^\smile, \circ\}$ with corresponding arities 2, 1, 0, 0, 0, 1, and 2 that is isomorphic to some proper relation algebra is called *representable*. The class of representable relation algebras is denoted by RRA. Since every proper relation algebra and therefore also every representable relation algebra satisfies the axioms from the previous definition we have PA $\subseteq$ RRA $\subseteq$ RA. A classical result of Lyndon [34] states that there exist finite relation algebras $\mathbf{A} \in$ RA that are not representable; so the inclusions above are proper. If a relation algebra $\mathbf{A}$ is representable then the isomorphism to a proper relation algebra is usually called the *representation of* $\mathbf{A}$.

We will be interested in the model-theoretic behavior of sets of relations which form the domain of a proper relation algebra, and therefore consider relational structures whose relations are precisely the relations of a proper relation algebra. If the set of relations of a relational structure $\mathfrak{B}$ forms a proper relation algebra which is a representation of some abstract relation algebra $\mathbf{A}$, then it will be convenient to also call $\mathfrak{B}$ a representation of $\mathbf{A}$.

▶ **Definition 3.** *Let* $\mathbf{A} \in$ RA*. A* representation of $\mathbf{A}$ *is a relational structure* $\mathfrak{B}$ *such that*

- $\mathfrak{B}$ *is an* A*-structure, i.e., the elements of* A *are binary relation symbols of* $\mathfrak{B}$*;*
- *The map* $a \mapsto a^{\mathfrak{B}}$ *is an isomorphism between the abstract relation algebra* $\mathbf{A}$ *and the proper relation algebra* $(\mathcal{R}; \cup, ^-, 0, 1, \mathrm{Id}, ^\smile, \circ)$ *with domain* $\mathcal{R} := \{a^{\mathfrak{B}} \mid a \in A\}$*.*

Recall that the set of atoms of a relation algebra $\mathbf{A} = (A; \cup, ^-, 0, 1, \mathrm{Id}, ^\smile, \circ)$ is denoted by A_0. The following definitions are crucial for this article.

▶ **Definition 4.** *A tuple* $(x, y, z) \in (A_0)^3$ *is called an* allowed triple (of $\mathbf{A}$) *if* $z \leq x \circ y$*. Otherwise,* (x, y, z) *is called a* forbidden triple (of $\mathbf{A}$)*; in this case* $\overline{z} \cup \overline{x \circ y} = 1$*. We say that a relational* A*-structure* $\mathfrak{B}$ induces a forbidden triple (from $\mathbf{A}$) *if there exist* $b_1, b_2, b_3 \in B$ *and* $(x, y, z) \in (A_0)^3$ *such that* $x(b_1, b_2), y(b_2, b_3)$ *and* $z(b_1, b_3)$ *hold in* $\mathfrak{B}$ *and* (x, y, z) *is a forbidden triple of* $\mathbf{A}$*.*

Note that a representation of $\mathbf{A}$ by definition does not induce a forbidden triple. A relation $R \subseteq A^3$ is called *totally symmetric* if for every bijection $\pi\colon \{1,2,3\} \to \{1,2,3\}$ we have

$$(a_1, a_2, a_3) \in R \;\Rightarrow\; (a_{\pi(1)}, a_{\pi(2)}, a_{\pi(3)}) \in R.$$

The following is an immediate consequence of the definition of allowed triples.

▶ Remark 5. The set of allowed triples of a symmetric relation algebra $\mathbf{A}$ is totally symmetric.

2.2 The Network Satisfaction Problem

In this section we present computational decision problems associated with relation algebras. We first introduce the inputs to these decision problems, so-called $\mathbf{A}$-networks.

▶ **Definition 6.** *Let $\mathbf{A}$ be a relation algebra. An* $\mathbf{A}$-network $(V; f)$ *is a finite set V together with a partial function $f\colon E \subseteq V^2 \to A$, where E is the domain of f. An $\mathbf{A}$-network $(V; f)$ is* satisfiable in a representation $\mathfrak{B}$ *of $\mathbf{A}$ if there exists an assignment $s\colon V \to B$ such that for all $(x, y) \in E$ the following holds:*

$$(s(x), s(y)) \in f(x, y)^{\mathfrak{B}}.$$

An $\mathbf{A}$-network $(V; f)$ is satisfiable *if there exists a representation $\mathfrak{B}$ of $\mathbf{A}$ such that $(V; f)$ is satisfiable in $\mathfrak{B}$.*

With these notions we can define the network satisfaction problem.

▶ **Definition 7.** *The* (general) network satisfaction problem *for a finite relation algebra $\mathbf{A}$, denoted by* $\mathrm{NSP}(\mathbf{A})$*, is the problem of deciding whether a given $\mathbf{A}$-network is satisfiable.*

In the following we assume that for an $\mathbf{A}$-network $(V; f)$ it holds that $f(V^2) \subseteq A \setminus \{0\}$. Otherwise, $(V; f)$ is not satisfiable. Note that every $\mathbf{A}$-network $(V; f)$ can be viewed as an A-structure $\mathfrak{C}$ on the domain V: for all $x, y \in V$ in the domain of f and $a \in A$ the relation $a^{\mathfrak{C}}(x, y)$ holds if and only if $f(x, y) = a$.

It is well-known that for relation algebras $\mathbf{A}_1$ and $\mathbf{A}_2$ the direct product $\mathbf{A}_1 \times \mathbf{A}_2$ is also a relation algebra (see, e.g., [30]). We will see in Lemma 9 that the direct product of representable relation algebras is also a representable relation algebra.

▶ **Definition 8.** *Let $\mathbf{A}_1$ and $\mathbf{A}_2$ be representable relation algebras. Let $\mathfrak{B}_1$ and $\mathfrak{B}_2$ be representations of $\mathbf{A}_1$ and $\mathbf{A}_2$ with disjoint domains. Then the* union representation *of the direct product $\mathbf{A}_1 \times \mathbf{A}_2$ is the $(A_1 \times A_2)$-structure $\mathfrak{B}_1 \uplus \mathfrak{B}_2$ on the domain $B_1 \uplus B_2$ with the following definition for all $(a_1, a_2) \in A_1 \times A_2$:*

$$(a_1, a_2)^{\mathfrak{B}_1 \uplus \mathfrak{B}_2} := a_1^{\mathfrak{B}_1} \cup a_2^{\mathfrak{B}_2}.$$

The following well-known lemma establishes a connection between products of relation algebras and union representations (see, e.g., Lemma 7 in [21] or Lemma 3.7 in [30]); it states that union representations are indeed representations. A proof of the lemma can be found, for example, in the extended version of this article [13].

▶ **Lemma 9.** *Let $\mathbf{A}_1$ and $\mathbf{A}_2$ be relation algebras. Then the following holds:*

1. *If $\mathfrak{B}_1$ and $\mathfrak{B}_2$ are representations of $\mathbf{A}_1$ and $\mathbf{A}_2$ with disjoint domains, then $\mathfrak{B}_1 \uplus \mathfrak{B}_2$ is a representation of $\mathbf{A}_1 \times \mathbf{A}_2$.*
2. *If $\mathfrak{B}$ is a representation of $\mathbf{A}_1 \times \mathbf{A}_2$, then there exist representations $\mathfrak{B}_1$ and $\mathfrak{B}_2$ of $\mathbf{A}_1$ and $\mathbf{A}_2$ such that $\mathfrak{B}$ is isomorphic to $\mathfrak{B}_1 \uplus \mathfrak{B}_2$.*

The following result uses Lemma 9 to obtain reductions between different network satisfaction problems. A similar statement can be found in Lemma 7 from [21], however there the assumption on representability of the relation algebras $\mathbf{A}$ and $\mathbf{B}$ is missing. Note that without this assumption the statement is not longer true. Consider relation algebras $\mathbf{A}$ and $\mathbf{B}$ such that NSP($\mathbf{A}$) is undecidable and $\mathbf{B}$ does not have a representation. Then $\mathbf{A} \times \mathbf{B}$ does also not have a representation (see Lemma 9) and hence NSP($\mathbf{A} \times \mathbf{B}$) is trivial. We observe that the undecidable problem NSP($\mathbf{A}$) cannot have a polynomial-time reduction to the trivial problem NSP($\mathbf{A} \times \mathbf{B}$).

▶ **Lemma 10.** *Let* $\mathbf{A}, \mathbf{B} \in \mathrm{RRA}$ *be finite. Then there exists a polynomial-time reduction from* NSP($\mathbf{A}$) *to* NSP($\mathbf{A} \times \mathbf{B}$).

Proof. Consider the following polynomial-time reduction from NSP($\mathbf{A}$) to NSP($\mathbf{A} \times \mathbf{B}$). We map a given $\mathbf{A}$-network $(V; f)$ to the $(\mathbf{A} \times \mathbf{B})$-network $(V; f')$ where f' is defined by $f'(x, y) := (f(x, y), 0)$. This reduction can be computed in polynomial time.

▷ Claim 1. If $(V; f)$ is satisfiable then $(V; f')$ is also satisfiable. Let $\mathfrak{A}$ be a representation of $\mathbf{A}$ in which $(V; f)$ is satisfiable and let $\mathfrak{B}$ be an arbitrary representation of $\mathbf{B}$. By Lemma 9, the structure $\mathfrak{A} \uplus \mathfrak{B}$ is a representation of $\mathbf{A} \times \mathbf{B}$. Moreover, the definition of union representations (Definition 8) yields that the $(\mathbf{A} \times \mathbf{B})$-network $(V; f')$ is satisfiable in $\mathfrak{A} \uplus \mathfrak{B}$.

▷ Claim 2. If $(V; f')$ is satisfiable then $(V; f)$ is satisfiable. Assume that $(V; f')$ is satisfiable in some representation $\mathfrak{C}$ of $\mathbf{A} \times \mathbf{B}$. By item 2 in Lemma 9 we get that $\mathfrak{C}$ is isomorphic to $\mathfrak{A} \uplus \mathfrak{B}$, where $\mathfrak{A}$ and $\mathfrak{B}$ are representations of $\mathbf{A}$ and $\mathbf{B}$. It again follows from the definition of union representations that $(V; f)$ is satisfiable in the representation $\mathfrak{A}$ of $\mathbf{A}$.

This shows the correctness of the polynomial-time reduction from NSP($\mathbf{A}$) to NSP($\mathbf{A} \times \mathbf{B}$) and finishes the proof. ◀

2.3 Normal Representations and Constraint Satisfaction Problems

We consider a subclass of RRA introduced by Hirsch in 1996. For relation algebras $\mathbf{A}$ from this class, NSP($\mathbf{A}$) corresponds naturally to a constraint satisfaction problem. In the following let $\mathbf{A}$ be in RRA. We call an $\mathbf{A}$-network $(V; f)$ *closed* (transitively closed in the work by Hirsch [28]) if f is total and for all $x, y, z \in V$ it holds that

- $f(x, x) \leq \mathrm{Id}$,
- $f(x, y) = \breve{a}$ for $a = f(y, x)$,
- $f(x, z) \leq f(x, y) \circ f(y, z)$.

It is called *atomic* if the range of f only contains atoms from $\mathbf{A}$.

▶ **Definition 11** (from [27]). *Let* $\mathfrak{B}$ *be a representation of* $\mathbf{A}$*. Then* $\mathfrak{B}$ *is called*

- fully universal*, if every atomic closed* $\mathbf{A}$*-network is satisfiable in* $\mathfrak{B}$*;*
- square*, if* $1^{\mathfrak{B}} = B^2$*;*
- homogeneous*, if for every isomorphism between finite substructures of* $\mathfrak{B}$ *there exists an automorphism of* $\mathfrak{B}$ *that extends this isomorphism;*
- normal*, if it is fully universal, square and homogeneous.*

We now investigate the connection between NSP($\mathbf{A}$) for a finite relation algebra with a normal representation $\mathfrak{B}$ and constraint satisfaction problems. Let τ be a finite relational signature and let $\mathfrak{B}$ be a (finite or infinite) τ-structure. Then the *constraint satisfaction problem for* $\mathfrak{B}$, denoted by CSP($\mathfrak{B}$), is the computational problem of deciding whether a finite input structure $\mathfrak{A}$ has a homomorphism to $\mathfrak{B}$. The structure $\mathfrak{B}$ is called the template of CSP($\mathfrak{B}$).

$\circ$	Id	$<$	$>$
Id	Id	$<$	$>$
$<$	$<$	$<$	1
$>$	$>$	1	$>$

Figure 1 Multiplication table of the point algebra $\mathbf{P}$.

Consider the following translation which associates to each $\mathbf{A}$-network $(V; f)$ an A-structure $\mathfrak{C}$ as follows: the set V is the domain of $\mathfrak{C}$ and $(x, y) \in C^2$ is in a relation $a^{\mathfrak{C}}$ if and only if (x, y) is in the domain of f and $f(x, y) = a$ holds. For the other direction let $\mathfrak{C}$ be an A-structure with domain C and consider the $\mathbf{A}$-network $(C; f)$ with the following definition: for every $x, y \in C$, if (x, y) does not appear in any relation of $\mathfrak{C}$ we leave $f(x, y)$ undefined, otherwise let $a_1(x, y), \dots, a_n(x, y)$ be all atomic formulas that hold in $\mathfrak{C}$. We compute in $\mathbf{A}$ the element $a := a_1 \cap \dots \cap a_n$ and define $f(x, y) := a$.

The following theorem is based on the natural 1-to-1 correspondence between $\mathbf{A}$-networks and A-structures; it subsumes the connection between network satisfaction problems and constraint satisfaction problems.

▶ **Proposition 12** (Proposition 1.3.16 in [6], see also [7, 9]). *Let $\mathbf{A} \in \mathrm{RRA}$ be finite. Then the following holds:*

1. *$\mathbf{A}$ has a representation $\mathfrak{B}$ such that $\mathrm{NSP}(\mathbf{A})$ and $\mathrm{CSP}(\mathfrak{B})$ are the same problem up to the translation between $\mathbf{A}$-networks and A-structures.*
2. *If $\mathbf{A}$ has a normal representation $\mathfrak{B}$ the problems $\mathrm{NSP}(\mathbf{A})$ and $\mathrm{CSP}(\mathfrak{B})$ are the same up to the translation between $\mathbf{A}$-networks and A-structures.*

Usually, normal representations of relation algebras are infinite relational structures. This means that the transfer from NSPs to CSPs from Proposition 12 results in CSPs over infinite templates, as in the following example.

▶ **Example 13.** Consider the *point algebra* $\mathbf{P}$. The set of atoms of $\mathbf{P}$ is $P_0 = \{\mathrm{Id}, <, >\}$. The composition operation $\circ$ on the atoms is given by the multiplication table in Figure 1. The table completely determines the composition operation $\circ$ on all elements of $\mathbf{P}$. Note that the structure $\mathfrak{P} := (\mathbb{Q}; \emptyset, <, >, =, \leq, \geq, \neq, \mathbb{Q}^2)$ is the normal representation of $\mathbf{P}$ and therefore $\mathrm{NSP}(\mathbf{P})$ and $\mathrm{CSP}(\mathfrak{P})$ are the same problems up to the translation between networks and structures.

2.4 The Universal-Algebraic Approach

In this section we give a brief introduction to the the universal-algebraic approach to CSPs.

2.4.1 Polymorphisms

Let τ be a finite relational signature. A *polymorphism* of a τ-structure $\mathfrak{B}$ is a homomorphism f from $\mathfrak{B}^k$ to $\mathfrak{B}$, for some $k \in \mathbb{N}$ called the *arity of f*. We write $\mathrm{Pol}(\mathfrak{B})$ for the set of all polymorphisms of $\mathfrak{B}$. The set of polymorphisms is closed under composition, i.e., for all n-ary $f \in \mathrm{Pol}(\mathfrak{B})$ and s-ary $g_1, \dots, g_n \in \mathrm{Pol}(\mathfrak{B})$ it holds that $f(g_1, \dots, g_n) \in \mathrm{Pol}(\mathfrak{B})$, where $f(g_1, \dots, g_n)$ is a homomorphism from $\mathfrak{B}^s$ to $\mathfrak{B}$ defined as follows

$$f(g_1, \dots, g_n)(x_1, \dots, x_s) := f(g_1(x_1, \dots, x_s), \dots, g_n(x_1, \dots, x_s)).$$

If $r_1, \dots, r_n \in B^k$ and $f\colon B^n \to B$ an n-ary operation, then we write $f(r_1, \dots, r_n)$ for the k-tuple obtained by applying f component-wise to the tuples $r_1, \dots, r_n$. We say that $f\colon B^n \to B$ *preserves* a k-ary relation $R \subseteq B^k$ if for all $r_1, \dots, r_n \in R$ it holds that $f(r_1, \dots, r_n) \in R$. We want to remark that the polymorphisms of $\mathfrak{B}$ are precisely those operations that preserve all relations from $\mathfrak{B}$.

A first-order τ-formula $\varphi(x_1, \dots, x_n)$ is called *primitive positive (pp)* if it has the form

$$\exists x_{n+1}, \dots, x_m(\varphi_1 \wedge \dots \wedge \varphi_s)$$

where $\varphi_1, \dots, \varphi_s$ are atomic τ-formulas, i.e., formulas of the form $R(y_1, \dots, y_l)$ for $R \in \tau$ and $y_i \in \{x_1, \dots, x_m\}$, of the form $y = y'$ for $y, y' \in \{x_1, \dots, x_m\}$, or of the form $\perp$. We say that a relation R is *primitively positively definable over* $\mathfrak{A}$ if there exists a primitive positive τ-formula $\varphi(x_1, \dots, x_n)$ such that R is definable over $\mathfrak{A}$ by $\varphi(x_1, \dots, x_n)$.

▶ **Proposition 14** ([15, 24]). *Let $\mathfrak{B}$ be a τ-structure with a finite domain. Then the set of primitive positive definable relations in $\mathfrak{B}$ is exactly the set of relations preserved by* $\mathrm{Pol}(\mathfrak{B})$.

2.4.2 Atom Structures

In this section we introduce for every finite $\mathbf{A} \in \mathrm{RA}$ an associated finite structure, called the *atom structure* of $\mathbf{A}$. If $\mathbf{A}$ has a fully universal representation, then there exists a polynomial-time reduction from $\mathrm{NSP}(\mathbf{A})$ to the finite-domain constraint satisfaction problem $\mathrm{CSP}(\mathfrak{A}_0)$ (Proposition 16). Hence, this reduction provides polynomial-time algorithms to solve NSPs, whenever the CSP of the associated atom structure can be solved in polynomial-time. For a discussion of the atom structure and related objects we recommend Section 4 in [12].

▶ **Definition 15.** *The* atom structure of $\mathbf{A} \in \mathrm{RA}$ *is the finite relational structure $\mathfrak{A}_0$ with domain A_0 and the following relations:*

- *for every $x \in A$ the unary relation $x^{\mathfrak{A}_0} := \{a \in A_0 \mid a \leq x\}$,*
- *the binary relation $E^{\mathfrak{A}_0} := \{(a_1, a_2) \in A_0^2 \mid \breve{a}_1 = a_2\}$,*
- *the ternary relation $R^{\mathfrak{A}_0} := \{(a_1, a_2, a_3) \in A_0^3 \mid a_3 \leq a_1 \circ a_2\}$.*

Note that $\mathfrak{A}_0$ has all subsets of A_0 as unary relations and that the relation $R^{\mathfrak{A}_0}$ consists of the allowed triples of $\mathbf{A} \in \mathrm{RRA}$. We say that an operation *preserves the allowed triples* if it preserves the relation $R^{\mathfrak{A}_0}$.

▶ **Proposition 16** ([11, 12]). *Let $\mathfrak{B}$ be a fully universal representation of a finite $\mathbf{A} \in \mathrm{RRA}$. Then there is a polynomial-time reduction from $\mathrm{CSP}(\mathfrak{B})$ to $\mathrm{CSP}(\mathfrak{A}_0)$.*

2.4.3 Conservative Clones

Let $\mathfrak{B}$ be a finite τ-structure. An operation $f\colon B^n \to B$ is called *conservative* if for all $x_1, \dots, x_n \in B$ it holds that $f(x_1, \dots, x_n) \in \{x_1, \dots, x_n\}$. The operation clone $\mathrm{Pol}(\mathfrak{B})$ is *conservative* if every $f \in \mathrm{Pol}(\mathfrak{B})$ is conservative. We call a relational structure $\mathfrak{B}$ *conservative* if $\mathrm{Pol}(\mathfrak{B})$ is conservative.

▶ Remark 17. Let $\mathfrak{A}_0$ be the atom structure of a finite relation algebra $\mathbf{A}$. Every $f \in \mathrm{Pol}(\mathfrak{A}_0)$ preserves all subsets of A_0, and is therefore conservative. Hence, $\mathrm{Pol}(\mathfrak{A}_0)$ is conservative.

This remark justifies our interest in the computational complexity of certain CSPs where the template has conservative polymorphisms. Their complexity can be studied via universal algebraic methods as we will see in the following. An operation $f\colon B^3 \to B$ is called

- a *majority operation* if $\forall x, y \in B. f(x, x, y) = f(x, y, x) = f(y, x, x) = x$;
- a *minority operation* if $\forall x, y \in B. f(x, x, y) = f(x, y, x) = f(y, x, x) = y$.

An operation $f\colon B^n \to B$, for $n \geq 2$, is called

- a *cyclic operation* if $\forall x_1,\dots,x_n \in B. f(x_1,\dots,x_n) = f(x_n,x_1,\dots,x_{n-1})$;
- a *weak near-unanimity operation* if

$$\forall x,y \in B. f(x,\dots,x,y) = f(x,\dots,x,y,x) = \dots = f(y,x,\dots,x);$$

- a *Siggers operation* if $n = 6$ and $\forall x,y \in B. f(x,x,y,y,z,z) = f(y,z,x,z,x,y)$.

The following terminology was introduced by Bulatov and has proven to be extremely powerful, especially in the context of conservative clones.

▶ **Definition 18** ([16, 17]). *A pair $(a,b) \in B^2$ is called a* semilattice edge *if there exists $f \in \mathrm{Pol}(\mathfrak{B})$ of arity two such that $f(a,b) = b = f(b,a) = f(b,b)$ and $f(a,a) = a$. A two-element set $\{a,b\} \subseteq B$* has a semilattice edge *if (a,b) or (b,a) is a semilattice edge.*

A two-element subset $\{a,b\}$ of B is called a majority edge *if neither (a,b) nor (b,a) is a semilattice edge and there exists an $f \in \mathrm{Pol}(\mathfrak{B})$ of arity three whose restriction to $\{a,b\}$ is a majority operation.*

A two-element subset $\{a,b\}$ of B is called an affine edge *if it is not a majority edge, if neither (a,b) nor (b,a) is a semilattice edge, and there exists an $f \in \mathrm{Pol}(\mathfrak{B})$ of arity three whose restriction to $\{a,b\}$ is a minority operation.*

If $S \subseteq B$ and $(a,b) \in S^2$ is a semilattice edge then we say that (a,b) *is a semilattice edge on S*. Similarly, if $\{a,b\} \subseteq S$ is a majority edge (affine edge) then we say that $\{a,b\}$ is a *majority edge on S* (*affine edge on S*).

The main result about conservative finite structures and their CSPs is the following dichotomy, first proved by Bulatov, 14 years before the proof of the Feder-Vardi conjecture.

▶ **Theorem 19** ([16]; see also [3, 17, 18]). *Let $\mathfrak{B}$ be a finite structure with a finite relational signature such that $\mathrm{Pol}(\mathfrak{B})$ is conservative. Then precisely one of the following holds:*

1. *$\mathrm{Pol}(\mathfrak{B})$ contains a Siggers operation; in this case, $\mathrm{CSP}(\mathfrak{B})$ is in P.*
2. *There exist distinct $a,b \in B$ such that for every $f \in \mathrm{Pol}(\mathfrak{B})^{(n)}$ the restriction of f to $\{a,b\}^n$ is a projection. In this case, $\mathrm{CSP}(\mathfrak{B})$ is NP-complete.*

Note that this means that $\mathrm{Pol}(\mathfrak{B})$ contains a Siggers operation if and only if for all two elements $a,b \in B$ the set $\{a,b\}$ is a majority edge, an affine edge, or there is a semilattice edge on $\{a,b\}$.

2.5 The k-Consistency Procedure

We present in the following the k-consistency procedure. It was introduced in [2] for finite structures and extended to infinite structures in several equivalent ways, for example in terms of Datalog programs, existential pebble games, and finite variable logics [8]. Also see [39] for recent results about the power of k-consistency for infinite-domain CSPs.

Let τ be a finite relational signature and let $k,l \in \mathbb{N}$ with $k < l$ and let $\mathfrak{B}$ be a fixed τ-structures with finitely many orbits of l-tuples. We define $\mathfrak{B}'$ to be the expansion of $\mathfrak{B}$ by all orbits of n-tuples for every $n \leq l$. We denote the extended signature of $\mathfrak{B}'$ by τ'. Let $\mathfrak{A}$ be an arbitrary finite τ-structure. A *partial l-decoration of* $\mathfrak{A}$ is a set g of atomic τ'-formulas such that

1. the variables of the formulas from g are a subset of A and denoted by $\mathrm{Var}(g)$,
2. $|\mathrm{Var}(g)| \leq l$,
3. the τ-formulas in g hold in $\mathfrak{A}$, where variables are interpreted as domain elements of $\mathfrak{A}$,
4. the conjunction over all formulas in g is satisfiable in $\mathfrak{B}'$.

A partial l-decoration g of $\mathfrak{A}$ is called *maximal* if there exists no partial l-decoration h of $\mathfrak{A}$ with $\mathrm{Var}(g) = \mathrm{Var}(h)$ such that $g \subsetneq h$. We denote the set of maximal partial l-decorations of $\mathfrak{A}$ by $\mathcal{R}^l_{\mathfrak{A}}$. Note that a fixed finite set of at most l variables, there are only finitely many partial l-decorations of $\mathfrak{A}$, because $\mathfrak{B}$ has by assumption finitely many orbits of l-tuples. Since this set is constant and can be precomputed, the set $\mathcal{R}^l_{\mathfrak{A}}$ can be computed efficiently. Then the (k,l)*-consistency procedure for* $\mathfrak{B}$ is the following algorithm.

Algorithm 1 (k,l)-consistency procedure for $\mathfrak{B}$.

Input : A finite τ-structure $\mathfrak{A}$.
1 **compute** $\mathcal{H} := \mathcal{R}^l_{\mathfrak{A}}$.
2 **repeat**
3 For every $f \in \mathcal{H}$ with $\mathrm{Var}(f) \leq k$ and every $U \subseteq A$ with $|U| \leq l-k$, if there does not exist $g \in \mathcal{H}$ with $f \subseteq g$ and $U \subseteq \mathrm{Dom}(g)$, then remove f from $\mathcal{H}$.
4 **until** $\mathcal{H}$ *does not change*
5 **if** $\mathcal{H}$ *is empty* **then**
6 **return** Reject.
7 **else**
8 **return** Accept.

Since $\mathcal{R}^l_{\mathfrak{A}}$ is of polynomial size (in the size of A) and the (k,l)-consistency procedure removes in step 3. at least one element from $\mathcal{R}^l_{\mathfrak{A}}$ the algorithm has a polynomial run time. The $(k,k+1)$-consistency procedure is also called *k-consistency procedure*. The $(2,3)$-consistency procedure is called *path consistency procedure*.[1]

▶ **Definition 20.** *Let $\mathfrak{B}$ be a relation τ-structure as defined before. Then the (k,l)*-consistency procedure for $\mathfrak{B}$ solves $\mathrm{CSP}(\mathfrak{B})$ *if the satisfiable instances of* $\mathrm{CSP}(\mathfrak{B})$ *are precisely the accepted instances of the (k,l)-consistency procedure.*

▶ Remark 21. Let $\mathbf{A}$ be a relation algebra with a normal representation $\mathfrak{B}$. We will in the following say that the k-consistency procedure solves $\mathrm{NSP}(\mathbf{A})$ if it solves $\mathrm{CSP}(\mathfrak{B})$. This definition is justified by the correspondence of NSPs and CSPs from Theorem 12.

▶ **Theorem 22** ([33]). *Let $\mathfrak{B}$ be a finite τ-structure. Then the following are equivalent:*

1. *There exist $k \in \mathbb{N}$ such that the k-consistency procedure solves* $\mathrm{CSP}(\mathfrak{B})$.
2. *$\mathfrak{B}$ has a 3-ary weak near-unanimity polymorphism f and a 4-ary weak near-unanimity polymorphism g such that: $\forall x,y,z \in B.\ f(y,x,x) = g(y,x,x,x)$.*

Let $\mathfrak{A}_0$ be the atom structure of a relation algebra $\mathbf{A}$ with a normal representation $\mathfrak{B}$. We finish this section by connecting the solvability of $\mathrm{CSP}(\mathfrak{A}_0)$ by k-consistency (or its characterization in terms of polymorphims from the previous proposition) with the solvability of $\mathrm{CSP}(\mathfrak{B})$ by k-consistency. By Remark 21 this gives a criterion for the solvability of $\mathrm{NSP}(\mathbf{A})$ by the k-consistency procedure.

The following theorem is from [39] building on ideas from [14]. We present it here in a specific formulation that already incorporates a correspondence between polymorphisms of the atom structure and canonical operations. For more details see [11, 12].

[1] Some authors also call it the *strong path consistency algorithm*, because some forms of the definition of the path consistency procedure are only equivalent to our definition of the path consistency procedure if $\mathfrak{B}$ has a transitive automorphism group.

▶ **Theorem 23** ([39]). *Let $\mathfrak{B}$ be a normal representation of a finite relation algebra* $\mathbf{A}$ *and* $\mathfrak{A}_0$ *the atom structure* $\mathbf{A}$*. If* $\mathrm{Pol}(\mathfrak{A}_0)$ *contains a 3-ary weak near-unanimity polymorphism f and a 4-ary weak near-unanimity polymorphism g such that* $\forall x, y, z \in B.\ f(y, x, x) = g(y, x, x, x)$*, then* $\mathrm{NSP}(\mathbf{A})$ *is solved by the* $(4, 6)$*-consistency algorithm.*

3 The Undecidability of RBCP, CON, and PC

In order to view RBCP as a decision problem, we need the following definitions. Let FRA be the set of all relation algebras $\mathbf{A}$ with domain $\mathcal{P}(\{1, \dots, n\})$.

▶ **Definition 24** (RBCP). *We define the following subsets of* FRA*:*
- RBCP *denotes the set such that* $\mathrm{NSP}(\mathbf{A})$ *is in P.*
- RBCP^c *denotes* $\mathrm{FRA} \setminus \mathrm{RBCP}$*.*
- CON *denotes the set such that* $\mathrm{NSP}(\mathbf{A})$ *is solved by k-consistency for some* $k \in \mathbb{N}$*.*
- PC *denotes the set such that* $\mathrm{NSP}(\mathbf{A})$ *is solved by path consistency.*

The following theorem is our first result. Note that this can be seen as a negative answer to Hirsch's Really Big Complexity Problem [27].

▶ **Theorem 25.** RBCP *is undecidable,* CON *is undecidable, and* PC *is undecidable.*

In our undecidability proofs we reduce from the following well-known undecidable problem for relation algebras [25].

▶ **Definition 26** (Rep). *Let* Rep *be the computational problem of deciding for a given* $\mathbf{A} \in \mathrm{FRA}$ *whether* $\mathbf{A}$ *has a representation.*

In our proof we also use the fact that there exists $\mathbf{U} \in \mathrm{FRA}$ such that $\mathrm{NSP}(\mathbf{U})$ is undecidable [29]. Note that $\mathbf{U} \in \mathrm{Rep}$ since the network satisfaction problem for non-representable relation algebras is trivial and therefore decidable.

Proof of Theorem 25. We reduce the problem Rep to RBCP^c. Consider the following reduction $f\colon \mathrm{FRA} \to \mathrm{FRA}$. For a given $\mathbf{A} \in \mathrm{FRA}$, we define $f(\mathbf{A}) := \mathbf{A} \times \mathbf{U}$.

▷ Claim 1. If $\mathbf{A} \in \mathrm{Rep}$ then $f(\mathbf{A}) \in \mathrm{RBCP}^c$. If $\mathbf{A}$ is representable, then $\mathbf{A} \times \mathbf{U}$ is representable by the first part of Lemma 9. Then there is a polynomial-time reduction from $\mathrm{NSP}(\mathbf{U})$ to $\mathrm{NSP}(\mathbf{A} \times \mathbf{U})$ by Lemma 10. This shows that $\mathrm{NSP}(\mathbf{A} \times \mathbf{U})$ is undecidable, and hence $f(\mathbf{A})$ is in RBCP^c.

▷ Claim 2. If $\mathbf{A} \in \mathrm{FRA} \setminus \mathrm{Rep}$ then $f(\mathbf{A}) \in \mathrm{RBCP}$. If $\mathbf{A}$ is not representable, then $\mathbf{A} \times \mathbf{U}$ is not representable by the second part of Lemma 9, and hence $\mathrm{NSP}(\mathbf{A} \times \mathbf{U})$ is trivial and in P, and therefore in RBCP.

Clearly, f is computable (even in polynomial time). Since Rep is undecidable [25], this shows that RBCP^c, and hence RBCP, is undecidable as well. The proof for CON and PC is analogous; all we need is the fact that $\mathrm{NSP}(\mathbf{U}) \notin \mathrm{CON}$ and $\mathrm{NSP}(\mathbf{U}) \notin \mathrm{PC}$. ◀

4 Tractability via k-Consistency

We provide in this section a criterion that ensures solvability of NSPs by the k-consistency procedure (Theorem 30). A relation algebra $\mathbf{A}$ is called *symmetric* if all its elements are symmetric, i.e., $\breve{a} = a$ for every $a \in A$. We will see in the following that the assumption on $\mathbf{A}$ to be symmetric will simplify the atom structure A_0 of $\mathbf{A}$, which has some advantages in the upcoming arguments.

▶ **Definition 27.** *Let* $\mathbf{A}$ *be a finite symmetric relation algebra with set of atoms* A_0*. We say that* $\mathbf{A}$ *admits a* Siggers behavior *if there exists an operation* $s\colon A_0^6 \to A_0$ *such that*

1. s *preserves the allowed triples of* $\mathbf{A}$*,*
2. $\forall x_1, \dots, x_6 \in A_0.\ s(x_1, \dots, x_6) \in \{x_1, \dots, x_6\}$*,*
3. s *satisfies the Siggers identity:* $\forall x, y, z \in A_0.\ s(x, x, y, y, z, z) = s(y, z, x, z, x, y)$*.*

▶ Remark 28. We mention that if $\mathbf{A}$ has a normal representation $\mathfrak{B}$, then $\mathbf{A}$ admits a Siggers behavior if and only if $\mathfrak{B}$ has a pseudo-Siggers polymorphism which is canonical with respect to $\mathrm{Aut}(\mathfrak{B})$; see [14].

We say that a finite symmetric relation algebra $\mathbf{A}$ *has all* 1*-cycles* if for every $a \in A_0$ the triple (a, a, a) is allowed. Details on the notion of cycles from the relation algebra perspective can be found in [36]. The relevance of the existence of 1-cycles for constraint satisfaction comes from the following observation.

▶ **Lemma 29.** *Let* $\mathbf{A}$ *be a finite symmetric relation algebra with a normal representation* $\mathfrak{B}$ *that has a binary injective polymorphism. Then* $\mathbf{A}$ *has all* 1*-cycles.*

Proof. Let i be a binary injective polymorphism of $\mathfrak{B}$ and let $a \in A_0$ be arbitrary. Consider $x_1, x_2, y_1, y_2 \in B$ such that $a^{\mathfrak{B}}(x_1, x_2)$ and $a^{\mathfrak{B}}(y_1, y_2)$. The application of i on the tuples (x_1, x_1, x_2) and (y_1, y_2, y_2) results in a substructure of $\mathfrak{B}$ that witnesses that (a, a, a) is an allowed triple. ◀

▶ **Theorem 30.** *Let* $\mathbf{A}$ *be a finite symmetric relation algebra with a normal representation* $\mathfrak{B}$*. Suppose that the following holds:*

1. $\mathbf{A}$ *has all* 1*-cycles.*
2. $\mathbf{A}$ *admits a Siggers behavior.*

Then the $\mathrm{NSP}(\mathbf{A})$ *can be solved by the* $(4, 6)$*-consistency procedure.*

We will outline the proof of Theorem 30 and cite some results from the literature that we will use. Assume that $\mathbf{A}$ is a finite symmetric relation algebra that satisfies the assumptions of Theorem 30. Since $\mathbf{A}$ admits a Siggers behavior there exists an operation $s\colon A_0^6 \to A_0$ that is by 1. and 2. in Definition 27 a polymorphism of the atom structure $\mathfrak{A}_0$ (see Paragraph 2.4.2). By Remark 17, $\mathrm{Pol}(\mathfrak{A}_0)$ is a conservative operation clone. Recall the notion of semilattice, majority, and affine edges for conservative clones (cf. Definition 18). Since s is by 3. a Siggers operation, Theorem 19 implies that every edge in $\mathfrak{A}_0$ is semilattice, majority, or affine.

Our goal is to show that there are no affine edges in $\mathfrak{A}_0$, since this implies that there exists $k \in \mathbb{N}$ such that $\mathrm{CSP}(\mathfrak{A}_0)$ can be solved by k-consistency [17]. We present this fact here via the characterization of (k, l)-consistency in terms of weak near-unanimity polymorphisms from Theorem 22.

▶ **Proposition 31** (cf. Corollary 3.2 in [31])**.** *Let* $\mathfrak{A}_0$ *be a finite conservative relational structure with a Siggers polymorphism and no affine edge. Then* $\mathfrak{A}_0$ *has a* 3*-ary weak near-unanimity polymorphism* f *and a* 4*-ary weak near-unanimity polymorphism* g *such that*

$$\forall x, y, z \in B.\ f(y, x, x) = g(y, x, x, x).$$

Note that the existence of the weak near-unanimity polymorphisms from Proposition 31 would finish the proof of Theorem 30, because Theorem 23 implies that in this case $\mathrm{NSP}(\mathbf{A})$ can be solved by the $(4, 6)$-consistency procedure. We therefore want to prove that there are no affine edges in $\mathfrak{A}_0$. We start by analyzing the different types of edges in the atom structure $\mathfrak{A}_0$ and obtain results about their appearance (see Section 4.1).

Fortunately, there is the following result by Alexandr Kazda about binary structures with a conservative polymophism clone. A *binary structure* is a structure where all relations have arity at most two.

▶ **Theorem 32** (Theorem 4.5 in [31]). *If $\mathfrak{A}$ is a finite binary conservative relational structure with a Siggers polymorphism, then $\mathfrak{A}$ has no affine edges.*

Notice that we cannot simply apply this theorem to the atom structure $\mathfrak{A}_0$, since the maximal arity of its relations is three. We circumvent this obstacle by defining for $\mathfrak{A}_0$ a closely related binary structure $\mathfrak{A}_0^{\mathrm{b}}$, which we call the "binarisation of $\mathfrak{A}_0$":

▶ **Definition 33.** *We denote by $\mathfrak{A}_0^{\mathrm{b}}$ the structure with domain A_0 and the following relations:*
- *a unary relation U_S for each subset S of A_0;*
- *for every $a \in A_0$ the binary relation $R_a := \{(x,y) \in A_0^2 \mid (a,x,y) \in R\}$;*
- *a relation for every union of relations of the form R_a.*

In the next step we investigate how $\mathrm{Pol}(\mathfrak{A}_0)$ and $\mathrm{Pol}(\mathfrak{A}_0^{\mathrm{b}})$ relate to each other. It follows from these observations that $\mathfrak{A}_0^{\mathrm{b}}$ does not have an affine edge. In other words, it only has semilattice and majority edges. The crucial step in our proof is to transfer a witness of this fact to $\mathfrak{A}_0$ and conclude that also $\mathfrak{A}_0$ has no affine edge. The detailed proofs can be found in the extended version of the article [13] and in the PhD thesis of the second author [32].

4.1 Results about the Atom Structure

In this section we present some of our findings about the atom structure. We obtain conditions on the atom structure (namely the ternary relation R) that imply the (non-)existence of semilatice edges in the atom structure. As we explained in the previous section, these results are the starting point for our proof of Theorem 30.

For the sake of notation, we make some global assumptions for this section. Let $\mathbf{A}$ be a finite relation algebra that satisfies the assumptions from Theorem 30. We denote by $\mathfrak{A}_0$ the atom structure of $\mathbf{A}$ (Definition 15). Since $\mathbf{A}$ is a symmetric relation algebra, the relation $R^{\mathfrak{A}_0}$ is totally symmetric. Furthermore, we can drop the binary relation $E^{\mathfrak{A}_0}$, since it consists only of loops and does not change the set of polymorphisms. Let $s \in \mathrm{Pol}(\mathfrak{A}_0)$ be the Siggers operation that exists by the assumptions in Theorem 30. This implies by Theorem 19 for every $a, b \in A_0$ that the set $\{a,b\}$ is a majority edge or an affine edge, or that there is a semilattice edge on $\{a,b\}$. The different types of edges are witnessed by certain operations that we get from Proposition 3.1.in [17]: there exist a binary operation $f \in \mathrm{Pol}(\mathfrak{A}_0)$ and ternary operations $g, h \in \mathrm{Pol}(\mathfrak{A}_0)$ such that for every two element subset C of A_0,
- $f|_C$ is a semilattice operation if C has a semilattice edge, and $f|_C(x,y) = x$ otherwise;
- $g|_C$ is a majority operation if C is a majority edge, $g|_C(x,y,z) = x$ if C is affine and $g|_C(x,y,z) = f|_C(f|_C(x,y),z)$ if C has a semilattice edge;
- $h|_C$ is a minority operation if C is an affine edge, $h|_C(x,y,z) = x$ if C is majority and $h|_B(x,y,z) = f|_C(f|_C(x,y),z)$ if C has a semilattice edge.

We will fix these operations and introduce the following terminology. A tuple $(a,b) \in A_0$ is called f-*sl* if $f(a,b) = b = f(b,a)$ holds. Next, we prove several important properties of the relation R: that it must contain certain triples (Lemma 34), that it must not contain certain other triples (Lemma 35), and that it is affected by the presence of semilattice edges in $\mathbf{A}_0$ (Lemma 36 and Lemma 37).

▶ **Lemma 34.** *The relation R of the atom structure $\mathfrak{A}_0$ has the following properties:*
- *for all $a \in A_0$ we have $(a,a,a) \in R$.*
- *for all $a, b \in A_0$ we have $(a,a,b) \in R$ or $(a,b,b) \in R$;*

Proof. The first item follows from the assumption that $\mathbf{A}$ has all 1-cycles.

For the second item observe that $\{a, \mathrm{Id}\}$ cannot be a majority edge. Otherwise,

$$g((a,a,\mathrm{Id}),(\mathrm{Id},a,a),(\mathrm{Id},\mathrm{Id},\mathrm{Id})) = (\mathrm{Id},a,\mathrm{Id}) \in R$$

is a contradiction to the properties of Id. Furthermore, (a, Id) cannot be f-sl, since

$$f((a,a,\mathrm{Id}),(\mathrm{Id},a,a)) = (\mathrm{Id},a,\mathrm{Id}) \in R.$$

This is again a contradiction. Since these observations also hold for b instead of a we have the following case distinction.

1. (Id, a) is f-sl and (Id, b) is f-sl. It follows that $f((a,a,\mathrm{Id}),(\mathrm{Id},b,b)) \in \{(a,a,b),(a,b,b)\}$. Since f preserves R, $(a,a,\mathrm{Id}) \in R$, and $(\mathrm{Id},b,b) \in R$ we get that $f((a,a,\mathrm{Id}),(\mathrm{Id},b,b)) \in R$. This implies that $(a,a,b) \in R$ or $(a,b,b) \in R$.
2. (Id, a) is f-sl and $\{b, \mathrm{Id}\}$ is affine. By the definition of f we get $f((b,b,\mathrm{Id}),(\mathrm{Id},a,a)) \in \{(b,a,a),(b,b,a)\}$. By the same argument as in Case 1 we get that $(a,a,b) \in R$ or $(a,b,b) \in R$.
3. (Id, b) is f-sl and $\{a, \mathrm{Id}\}$ is affine. This case is analogous to Case 2.
4. $\{a, \mathrm{Id}\}$ is affine and $\{b, \mathrm{Id}\}$ is affine. Observe that

   $$g((a,\mathrm{Id},a),(\mathrm{Id},b,b),(\mathrm{Id},\mathrm{Id},\mathrm{Id})) \in \{(a,b,a),(a,b,b)\},$$

   since $g(a,b,\mathrm{Id}) \in \{a,b,\mathrm{Id}\}$ and the triple (a,b,Id) is forbidden. As in the cases before it follows that $(a,a,b) \in R$ or $(a,b,b) \in R$.

This concludes the proof of the second item. ◀

▶ **Lemma 35.** *Let $a,b,c \in A_0$ be such that $(a,b,c) \notin R$ and $|\{a,b,c\}| = 3$. Then there are $x,y \in \{a,b,c\}$ such that $(x,x,y) \notin R$.*

Proof. We first suppose that there is a semilattice edge on $\{a,b,c\}$. Without loss of generality we assume that (a,b) is f-sl. If $f(c,a) = c$ then $(a,a,c) \notin R$ or $(b,a,a) \notin R$ because otherwise

$$f((a,a,c),(b,a,a)) = (b,a,c) \in R$$

contradicting our assumption. If $f(c,a) = a$ then $(b,c,c) \notin R$ or $(a,a,c) \notin R$ because otherwise

$$f((b,c,c),(a,a,c)) = (b,a,c) \in R$$

which is again a contradiction. Hence, in all the cases we found $x,y \in \{a,b,c\}$ such that $(x,x,y) \notin R$ and are done. In the following we therefore assume that there is no semilattice edge on $\{a,b,c\}$.

Next we suppose that there is an affine edge on $\{a,b,c\}$. Without loss of generality we assume that $\{a,b\}$ is an affine edge. Since there are no semilattice edges on $\{a,b,c\}$ we distinguish the following two cases:

1. $\{a,c\}$ is an affine edge. In this case $(c,a,a) \notin R$ or $(a,b,a) \notin R$ because otherwise

   $$h((c,a,a),(a,a,a),(a,b,a)) = (c,b,a) \in R.$$

2. $\{a,c\}$ is a majority edge. In this case $(a,a,c) \notin R$ or $(a,b,a) \notin R$ or $(b,b,c) \notin R$, because otherwise

   $$h((a,a,c),(a,b,a),(b,b,c)) = (b,a,c) \in R.$$

In both cases we again found $x, y \in \{a, b, c\}$ such that $(x, x, y) \notin R$ and are done. We therefore suppose in the following that there are no affine edges on $\{a, b, c\}$. Hence, all edges on $\{a, b, c\}$ are majority edges. Then $(a, a, c) \notin R$ or $(a, b, a) \notin R$ or $(b, b, c) \notin R$ because otherwise

$$g((a, a, c), (a, b, a), (b, b, c)) = (a, b, c) \in R.$$

Thus, also in this case we found $x, y \in \{a, b, c\}$ such that $(x, x, y) \notin R$. ◀

The next lemma states that the edge type on $\{a, b\}$ is predetermined whenever a triple (a, a, b) is not in R.

▶ **Lemma 36.** *Let $a, b \in A_0$ be such that $(a, a, b) \notin R$. Then (a, b) is a semilattice edge in $\mathfrak{A}_0$ but (b, a) is not.*

Proof. By Lemma 34 we know that $(a, b, b) \in R$, $(a, a, a) \in R$, and $(b, b, b) \in R$. Assume for contradiction that $\{a, b\}$ is a majority edge. Then

$$g((a, a, a), (a, b, b), (b, b, a)) = (a, b, a)$$

which contradicts the fact that g preserves R. Assume next that $\{a, b\}$ is an affine edge. Then

$$h((a, b, b), (b, a, b), (b, b, b)) = (a, a, b)$$

which again contradicts that h preserves R. Finally, if (b, a) is a semilattice edge then

$$f((a, b, b), (b, a, b)) = (a, a, b)$$

which contradicts the assumption that f preserves R. If follows that (a, b) is the only semilattice edge on $\{a, b\}$ and therefore $f(a, b) = b = f(b, a)$ holds. ◀

▶ **Lemma 37.** *Let $a, a', b, c \in A_0$ be such that $(a, b, c) \notin R$, $(a, a, b) \notin R$, and $(a', b, c) \in R$. Then (a', a) is not a semilattice edge.*

Proof. Assume for contradiction (a', a) is a semilattice edge, i.e., there exists $p \in \mathrm{Pol}(\mathfrak{A}_0)$ with $p(a, a') = a = p(a', a)$. Note that by Lemma 34 it follows that $(a, a, a) \in R$ and $(a, b, b) \in R$.

▷ Claim 1. $p(b, a) = a$ implies $p(a, b) = b$. This follows immediately, since otherwise $p((a, b, b), (b, a, b)) = (a, a, b) \in R$ is a contradiction.

▷ Claim 2. $(a, a, c) \notin R$. We assume the opposite and consider the only two possible cases for $p(b, a)$.

1. $p(b, a) = b$: We get a contradiction by $p((a', b, c), (a, a, c)) = (a, b, c) \in R$.
2. $p(b, a) = a$: By Claim 1 we know that $p(a, b) = b$ follows. Then $p((a, a, c), (a', b, c)) = (a, b, c) \in R$ contrary to our assumptions.

▷ Claim 3. $p(c, a) = a$ implies $p(a, c) = c$. Lemma 34 together with Claim 2 implies that $(a, c, c) \in R$. Now Claim 3 follows immediately, since otherwise $p((a, c, c), (c, a, c)) = (a, a, c) \in R$, which contradicts Claim 2.

We finally make a case distinction for all possible values of p on (b, a) and (c, a).

1. $p(b, a) = b$ and $p(c, a) = c$: We get a contradiction by $p((a', b, c), (a, a, a)) = (a, b, c) \in R$.
2. $p(b, a) = b$ and $p(c, a) = a$: We get a contradiction by $p((a', b, c), (a, a, a)) = (a, b, a) \in R$.
3. $p(b, a) = a$ and $p(c, a) = c$: $p((a', b, c), (a, a, a)) = (a, a, c) \in R$ contradicts Claim 2.
4. $p(b, a) = a$ and $p(c, a) = a$: By Claim 1 we get $p(a, b) = b$ and by Claim 3 we get $p(a, c) = c$. This yields a contradiction by $p((a, a, a), (a', b, c)) = (a, b, c) \in R$. ◀

5 k-Consistency and Symmetric Flexible-Atom Algebras

We apply our result from Section 4 to the class of finite symmetric relation algebras with a flexible atom and obtain a k-consistency versus NP-complete complexity dichotomy.

A finite relation algebra $\mathbf{A}$ is called *integral* if the element Id is an atom of $\mathbf{A}$, i.e., $\mathrm{Id} \in A_0$. We define flexible atoms for integral relation algebras only. For a discussion about integrality and flexible atoms consider Section 3 in [12].

▶ **Definition 38.** *Let $\mathbf{A} \in \mathrm{RA}$ be finite and integral. An atom $s \in A_0$ is called* flexible *if for all $a, b \in A \setminus \{\mathrm{Id}\}$ it holds that $s \leq a \circ b$.*

Relation algebras with a flexible atom have been studied intensively in the context of the *flexible atoms conjecture* [1, 35]. It can be shown easily that finite relation algebras with a flexible atom have a normal representation [11, 12]. In [12] the authors obtained a P versus NP-complete complexity dichotomy for NSPs of finite symmetric relation algebras with a flexible atom (assuming $\mathrm{P} \neq \mathrm{NP}$). In the following we strengthen this result and prove that every problem in this class can be solved by k-consistency for some $k \in \mathbb{N}$ or is NP-complete (without any complexity-theoretic assumptions).

We combine Theorem 30 with the main result of [12] to obtain the following characterization for NSPs of finite symmetric relation algebras with a flexible atom that are solved by the $(4, 6)$-consistency procedure. Note that the difference of Theorem 39 and the related result in [12] is the algorithm that solves the problems in P.

▶ **Theorem 39.** *Let $\mathbf{A}$ be a finite symmetric integral relation algebra with a flexible atom. Then the following are equivalent:*

- $\mathbf{A}$ *admits a Siggers behavior.*
- $\mathrm{NSP}(\mathbf{A})$ *can be solved by the $(4, 6)$-consistency procedure.*

Proof. Every finite symmetric relation algebra $\mathbf{A}$ with a flexible atom has a normal representation $\mathfrak{B}$ by Proposition 3.5 in [12].

If the first item holds it follows from Proposition 6.1. in [12] that $\mathfrak{B}$ has a binary injective polymorphism. By Lemma 29 the relation algebra $\mathbf{A}$ has all 1-cycles. We apply Theorem 30 and get that the second item in Theorem 39 holds.

We prove the converse implication by showing the contraposition. Assume that the first item is not satisfied. Then Theorem 9.1 in [12] implies that there exists a polynomial-time reduction from $\mathrm{CSP}(K_3)$ to $\mathrm{NSP}(\mathbf{A})$ which preserves solvability by the (k, l)-consistency procedure. The problem $\mathrm{CSP}(K_3)$ is the 3-colorability problem which is known (e.g., by [4]) to be not solvable by the (k, l)-consistency procedure for every $k, l \in \mathbb{N}$. Hence $\mathrm{NSP}(\mathbf{A})$ cannot be solved by the $(4, 6)$-consistency procedure. ◀

As a consequence of Theorem 39 we obtain the following strengthening of the complexity dichotomy NSPs of finite symmetric integral relation algebra with a flexible atom [12].

▶ **Corollary 40** (Complexity Dichotomy)**.** *Let $\mathbf{A}$ be a finite symmetric integral relation algebra with a flexible atom. Then $\mathrm{NSP}(\mathbf{A})$ can be solved by the $(4, 6)$-consistency procedure, or it is NP-complete.*

Proof. Suppose that the first condition in Theorem 39 holds. Then Theorem 39 implies that $\mathrm{NSP}(\mathbf{A})$ can be solved by the $(4, 6)$-consistency procedure. If the first condition in Theorem 39 is not satisfied it follows from Theorem 9.1. in [12] that $\mathrm{NSP}(\mathbf{A})$ is NP-complete. ◀

6 The Complexity of the Meta Problem

In this section we study the computational complexity of deciding for a given finite symmetric relation algebra $\mathbf{A}$ with a flexible atom whether the k-consistency algorithm solves $\mathrm{NSP}(\mathbf{A})$. We show that this problem is decidable in polynomial time even if $\mathbf{A}$ is given by the restriction of its composition table to the atoms of $\mathbf{A}$: note that this determines a symmetric relation algebra uniquely, and that this is an (exponentially) more succinct representation of $\mathbf{A}$ compared to explicitly storing the full composition table.

▶ **Definition 41** (Meta Problem). *We define* Meta *to be the following computational problem.*
Input: *the composition table of a finite symmetric relation algebra* $\mathbf{A}$ *restricted to* A_0*.*
Question: *is there a* $k \in \mathbb{N}$ *such that* k*-consistency solves* $\mathrm{NSP}(\mathbf{A})$*?*

Our proof of Theorem 25 shows that Meta is undecidable as well.

▶ **Theorem 42.** Meta *can be decided in polynomial time if the input is restricted to finite symmetric integral relation algebras* $\mathbf{A}$ *with a flexible atom.*

Proof. By Theorem 39 it suffices to test the existence of an operation $f\colon A_0^6 \to A_0$ which satisfies conditions 1.-3. in this theorem. The three conditions can clearly be checked in polynomial time, so we already know that Meta is in NP.

Note that the search for f may be phrased as an instance of $\mathrm{CSP}(\mathfrak{A}_0)$ with $|A|^6$ variables. Using the fact that the k-consistency procedure is one-sided correct even in the case that $\mathrm{CSP}(\mathfrak{A}_0)$ is NP-hard (i.e., if the procedure rejects a given instance of $\mathrm{CSP}(\mathfrak{A}_0)$, then the instance is always unsatisfiable), we may use a standard self-reducibility argument (see, e.g., [20]) to obtain a polynomial-time algorithm for finding f. ◀

7 Conclusion and Open Questions

The question whether the network satisfaction problem for a given finite relation algebra can be solved by the famous k-consistency procedure is undecidable. Our proof of this fact heavily relies on prior work of Hirsch [29] and of Hirsch and Hodkinson [25] and shows that almost any question about the network satisfaction problem for finite relation algebras is undecidable.

However, if we further restrict the class of finite relation algebras, one may obtain strong classification results. We have demonstrated this for the class of finite symmetric integral relation algebras with a flexible atom (Theorem 40); the complexity of deciding whether the conditions in our classification result hold drops from undecidable to P (Theorem 42). One of the remaining open problems is a characterisation of the power of k-consistency for the larger class of all finite relation algebras with a normal representation.

Our main result (Theorem 30) is a sufficient condition for the applicability of the k-consistency procedure; the condition does not require the existence of a flexible atom but applies more generally to finite symmetric relation algebras $\mathbf{A}$ with a normal representation. Our condition consists of two parts: the first is the existence of all 1-cycles in $\mathbf{A}$, the second is that $\mathbf{A}$ admits a Siggers behavior. We conjecture that dropping the first part of the condition leads to a necessary and sufficient condition for solvability by the k-consistency procedure.

▶ **Conjecture 43.** *A finite symmetric relation algebra* $\mathbf{A}$ *with a normal representation admits a Siggers behavior if and only if* $\mathrm{NSP}(\mathbf{A})$ *can be solved by the* k*-consistency procedure for some* $k \in \mathbb{N}$*.*

$\circ$	Id	E	N
Id	Id	E	N
E	E	Id	N
N	N	N	1

Figure 2 Multiplication table of the relation algebra $\mathbf{C}$.

Note that this conjecture generalises Theorem 39. Both directions of the conjecture are open. However, the forward direction of the conjecture is true if $\mathbf{A}$ has a normal representation with a primitive automorphism group: in this case, it is known that a Siggers behavior implies the existence of all 1-cycles [10], and hence the claim follows from our main result (Theorem 39). The following example shows a finite symmetric relation algebra $\mathbf{A}$ which does not have all 1-cycles and an imprimitive normal representation, but still NSP($\mathbf{A}$) can be solved by the k-consistency procedure for some $k \in \mathbb{N}$.

▶ **Example 44.** Theorem 30 is a sufficient condition for the NSP of a relation algebra $\mathbf{A}$ to be solved by the k-consistency procedure for some $k \in \mathbb{N}$. However, there exists a finite symmetric relation algebra $\mathbf{C}$ such that NSP($\mathbf{C}$) is solved by the 2-consistency procedure, but we cannot prove this by the methods used to obtain Theorem 30. Consider the relation algebra $\mathbf{C}$ with atoms $\{\mathrm{Id}, E, N\}$ and the multiplication table in Figure 2. This relation algebra has a normal representation, namely the expansion of the infinite disjoint union of the clique K_2 by all first-order definable binary relations. We denote this structure by $\overline{\omega K_2}$. One can observe that CSP($\overline{\omega K_2}$) and therefore also the NSP of the relation algebra can be solved by the $(2,3)$-consistency algorithm (for details see [32]).

The relation algebra $\mathbf{C}$ does not have all 1-cycles and therefore does not fall into the scope of Theorem 30. In fact, our proof of Theorem does not work for $\mathbf{C}$, because the CSP of the atom structure $\mathfrak{C}_0$ of $\mathbf{C}$ cannot be solved by the k-consistency procedure for some $k \in \mathbb{N}$. Hence, the reduction of NSP($\mathbf{C}$) to CSP($\mathfrak{C}_0$) (incorporated in Theorem 23) does not imply that NSP($\mathbf{C}$) can be solved by k-consistency procedure for some $k \in \mathbb{N}$.

The following problems are still open and are relevant for resolving Conjecture 43.

- Show Conjecture 43 if the normal representation of $\mathbf{A}$ has a primitive automorphism group.
- Characterise the power of the k-consistency procedure for the NSP of finite relation algebras with a normal representation whose automorphism group is imprimitive. In this case, there is a non-trivial definable equivalence relation. It is already known that if this equivalence relation has finitely many classes, then the NSP is NP-complete and the k-consistency procedure does not solve the NSP [10]. Similarly, the NSP is NP-complete if there are equivalence classes of finite size larger than two. It therefore remains to study the case of infinitely many two-element classes, and with infinitely many infinite classes. In both cases we wish to reduce the classification to the situation with a primitive automorphism group.

Finally, we ask whether it is true that if $\mathbf{A}$ is a finite symmetric relation algebra with a flexible atom and NSP($\mathbf{A}$) can be solved by the k-consistency procedure for some k, then it can also be solved by the $(2,3)$-consistency procedure? In other words, can we improve $(4,6)$ in Corollary 40 to $(2,3)$?

References

1 Jeremy F. Alm, Roger D. Maddux, and Jacob Manske. Chromatic graphs, Ramsey numbers and the flexible atom conjecture. *The Electronic Journal of Combinatorics*, 15(1), March 2008. doi:10.37236/773.

2 Albert Atserias, Andrei A. Bulatov, and Víctor Dalmau. On the power of k-consistency. In *ICALP*, pages 279–290, 2007.

3 Libor Barto. The dichotomy for conservative constraint satisfaction problems revisited. In *Proceedings of the Symposium on Logic in Computer Science (LICS)*, Toronto, Canada, 2011.

4 Libor Barto and Marcin Kozik. Constraint satisfaction problems of bounded width. In *Proceedings of Symposium on Foundations of Computer Science (FOCS)*, pages 595–603, 2009.

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

6 Manuel Bodirsky. Complexity classification in infinite-domain constraint satisfaction. Mémoire d'habilitation à diriger des recherches, Université Diderot – Paris 7. Available at arXiv:1201.0856v8, 2012.

7 Manuel Bodirsky. Finite relation algebras with normal representations. In *Relational and Algebraic Methods in Computer Science – 17th International Conference, RAMiCS 2018, Groningen, The Netherlands, October 29 – November 1, 2018, Proceedings*, pages 3–17, 2018.

8 Manuel Bodirsky and Víctor Dalmau. Datalog and constraint satisfaction with infinite templates. *Journal on Computer and System Sciences*, 79:79–100, 2013. A preliminary version appeared in the proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS'05).

9 Manuel Bodirsky and Peter Jonsson. A model-theoretic view on qualitative constraint reasoning. *Journal of Artificial Intelligence Research*, 58:339–385, 2017.

10 Manuel Bodirsky and Simon Knäuer. Hardness of network satisfaction for relation algebras with normal representations. In *Relational and Algebraic Methods in Computer Science*, pages 31–46. Springer International Publishing, 2020. doi:10.1007/978-3-030-43520-2_3.

11 Manuel Bodirsky and Simon Knäuer. Network satisfaction for symmetric relation algebras with a flexible atom. In *Proceedings of AAAI*, 2021. Preprint https://arxiv.org/abs/2008.11943.

12 Manuel Bodirsky and Simon Knäuer. The complexity of network satisfaction problems for symmetric relation algebras with a flexible atom. *Journal of Artificial Intelligence Research*, 75:1701–1744, December 2022. doi:10.1613/jair.1.14195.

13 Manuel Bodirsky and Simon Knäuer. Network satisfaction problems solved by k-consistency, 2023.

14 Manuel Bodirsky and Antoine Mottet. A dichotomy for first-order reducts of unary structures. *Logical Methods in Computer Science*, 14(2), 2018. doi:10.23638/LMCS-14(2:13)2018.

15 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.

16 Andrei A. Bulatov. Tractable conservative constraint satisfaction problems. In *Proceedings of the Symposium on Logic in Computer Science (LICS)*, pages 321–330, Ottawa, Canada, 2003.

17 Andrei A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Trans. Comput. Logic*, 12(4), July 2011. doi:10.1145/1970398.1970400.

18 Andrei A. Bulatov. Conservative constraint satisfaction re-revisited. *Journal Computer and System Sciences*, 82(2):347–356, 2016. ArXiv:1408.3690. doi:10.1016/j.jcss.2015.07.004.

19 Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17*, pages 319–330, 2017.

20 Hubie Chen and Benoît Larose. Asking the metaquestions in constraint tractability. *TOCT*, 9(3):11:1–11:27, 2017.

21 Matteo Cristiani and Robin Hirsch. The complexity of the constraint satisfaction problem for small relation algebras. *Artificial Intelligence Journal*, 156:177–196, 2004.

22 Ivo Düntsch. Relation algebras and their application in temporal and spatial reasoning. *Artificial Intelligence Review*, 23:315–357, 2005.

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

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

25 R. Hirsch and I. Hodkinson. Representability is not decidable for finite relation algebras. *Transactions of the American Mathematical Society*, 353(4):1387–1401), 2001.

26 R. Hirsch and I. Hodkinson. Strongly representable atom structures of relation algebras. *Transactions of the American Mathematical Society*, 130(6):1819–1831), 2001.

27 Robin Hirsch. Relation algebras of intervals. *Artificial Intelligence Journal*, 83:1–29, 1996.

28 Robin Hirsch. Expressive power and complexity in algebraic logic. *Journal of Logic and Computation*, 7(3):309–351, 1997.

29 Robin Hirsch. A finite relation algebra with undecidable network satisfaction problem. *Logic Journal of the IGPL*, 7(4):547–554, 1999.

30 Robin Hirsch and Ian Hodkinson. *Relation Algebras by Games*. North Holland, 2002.

31 Alexandr Kazda. CSP for binary conservative relational structures. *Algebra universalis*, 75(1):75–84, December 2015. doi:10.1007/s00012-015-0358-8.

32 Simon Knäuer. *Constraint Network Satisfaction for Finite Relation Algebras*. PhD thesis, Technische Universität Dresden, 2023.

33 Marcin Kozik, Andrei Krokhin, Matt Valeriote, and Ross Willard. Characterizations of several Maltsev conditions. *Algebra universalis*, 73(3):205–224, 2015. doi:10.1007/s00012-015-0327-2.

34 R. Lyndon. The representation of relational algebras. *Annals of Mathematics*, 51(3):707–729, 1950.

35 Roger D. Maddux. A perspective on the theory of relation algebras. *Algebra Universalis*, 31(3):456–465, September 1994. doi:10.1007/bf01221799.

36 Roger D. Maddux. *Relation Algebras: Volume 150*. Studies in logic and the foundations of mathematics. Elsevier Science, London, England, May 2006.

37 Ralph McKenzie. *The representation of relation algebras*. PhD thesis, University of Colorado at Boulder, 1966.

38 Ralph McKenzie. Representations of integral relation algebras. *Michigan Mathematical Journal*, 17(3):279–287, 1970. doi:10.1307/mmj/1029000477.

39 Antoine Mottet, Tomás Nagy, Michael Pinsker, and Michal Wrona. Smooth approximations and relational width collapses. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 138:1–138:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.138.

40 Jochen Renz and Bernhard Nebel. Qualitative spatial reasoning using constraint calculi. In M. Aiello, I. Pratt-Hartmann, and J. van Benthem, editors, *Handbook of Spatial Logics*, pages 161–215. Springer Verlag, Berlin, 2007.

41 Alfred Tarski. Representation problems for relation algebras. *Bulletin of the AMS*, 54(80), 1948.

42 Dmitriy Zhuk. A proof of the CSP dichotomy conjecture. *J. ACM*, 67(5):30:1–30:78, 2020. doi:10.1145/3402029.

43 Dmitriy N. Zhuk. A proof of CSP dichotomy conjecture. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17*, pages 331–342, 2017. https://arxiv.org/abs/1704.01914.