Constraint Network Satisfaction for Finite Relation Algebras

D I S S E R T A T I O N

zur Erlangung des akademischen Grades

Doctor rerum naturalium
(Dr. rer. nat.)

vorgelegt

dem Bereich Mathematik und Naturwissenschaften
der Technischen Universität Dresden

von

M.Sc. Simon Knäuer

Eingereicht am 20. Oktober 2022
Verteidigt am 03. April 2023

Die Dissertation wurde in der Zeit von Oktober 2018 bis Oktober 2022 am Institut für Algebra angefertigt.

Gutachter
Prof. Dr. Manuel Bodirsky, Technische Universität Dresden
Prof. Andrei A. Bulatov, Ph.D., Simon Fraser University

Abstract

Network satisfaction problems (*NSPs*) for finite relation algebras are computational decision problems, studied intensively since the 1990s. The major open research challenge in this field is to understand which of these problems are solvable by polynomial-time algorithms. Since there are known examples of undecidable NSPs of finite relation algebras it is advisable to restrict the scope of such a classification attempt to well-behaved subclasses of relation algebras. The class of relation algebras with a *normal representation* is such a well-behaved subclass. Many well-known examples of relation algebras, such as the Point Algebra, RCC5, and Allen's Interval Algebra admit a normal representation. The great advantage of finite relation algebras with normal representations is that their NSP is essentially the same as a *constraint satisfaction problem* (*CSP*). For a relational structure $\mathfrak{B}$ the problem $\mathrm{CSP}(\mathfrak{B})$ is the computational problem to decide whether a given finite relational structure $\mathfrak{C}$ has a homomorphism to $\mathfrak{B}$. The study of CSPs has a long and rich history, culminating for the time being in the celebrated proofs of the Feder-Vardi dichotomy conjecture. Bulatov and Zhuk independently proved that for every finite structure $\mathfrak{B}$ the problem $\mathrm{CSP}(\mathfrak{B})$ is in P or NP-complete. Both proofs rely on the *universal-algebraic approach*, a powerful theory that connects algebraic properties of structures $\mathfrak{B}$ with complexity results for the decision problems $\mathrm{CSP}(\mathfrak{B})$.

The CSPs that emerge from NSPs are typically of the form $\mathrm{CSP}(\mathfrak{B})$ for an infinite structure $\mathfrak{B}$ and therefore do not fall into the scope of the dichotomy result for finite structures. In this thesis we study NSPs of finite relation algebras with normal representations by the universal algebraic methods which were developed for the study of finite and infinite-domain CSPs. We additionally make use of model theory and a Ramsey-type result of Nešetřil and Rödl.

Our contributions to the field are divided into three parts. Firstly, we provide two algebraic criteria for NP-hardness of NSPs. Our second result is a complete classification of the complexity of NSPs for symmetric relation algebras with a flexible atom; these problems are in P or NP-complete. Our result is obtained via a decidable condition on the relation algebra which implies polynomial-time tractability of the NSP. As a third contribution we prove that for a large class of NSPs, non-hardness implies that the problems can even be solved by Datalog programs, unless $\mathrm{P} = \mathrm{NP}$. This result can be used to strengthen the dichotomy result for NSPs of symmetric relation algebras with a flexible atom: every such problem can be solved by a Datalog program or is NP-complete. Our proof relies equally on known results and new observations in the algebraic analysis of finite structures.

Acknowledgment

My time as a PhD student, was accompanied by many people. I would like to express my deep gratefulness for the support in the emergence of this work.

First of all, I would like to express my gratitude to my supervisor Professor Manuel Bodirsky. Thank you for introducing me to an incredibly fascinating and multifaceted field of research, for your always constructive guidance, the various open questions you shared with me, and the great confidence to work on them freely.

I would like to thank all my co-authors for the productive and instructive collaboration.

I thank the research training group QuantLA and Professor Franz Baader, who have funded me over the past four years. I really enjoyed to participate in QuantLA, to meet lots of interesting people and to learn about new topics.

It has been a great pleasure to be part of the CSP research community. The exchange with colleagues from Dresden, Prague, Vienna, and Hamburg was not only valuable in terms of content, but was often a great source of motivation. I will always keep the CSP World Congress 2020 in the best memory.

I would like to thank all the members of the Institute of Algebra for providing such a pleasant work environment. It has been a great enrichment for me to meet many different people over the past years and to talk about all kinds of things during lunch and coffee breaks.

I thank the secretaries of the Institute of Algebra and the Institute of Theoretical Computer Science for any administrative support.

Thank you, Albert, Florian, Jakub, and Leopold for all the mathematical and non-mathematical talks!

It is absolutely clear that apart from all this support, there is a whole other area where I have received invaluable support, for which I would like to thank both my family and my friends.

Finally, it is the three of you who I thank from the bottom of my heart!

Grant Agreement No 681988, CSP-Infinity

DFG Research Training Group 1763

The author was supported by DFG Graduiertenkolleg 1763 (QuantLA) and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 681988, "CSP-Infinity").

Contents

Chapter 1

Introduction

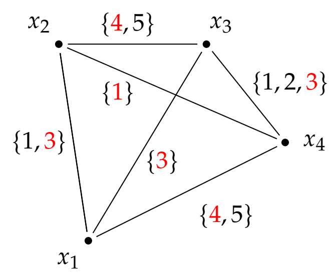
Until the middle of the 19th century, the study of formal logic in the West was predominantly limited to traditional logics without predicates and quantifiers. Beginning in the 1860s, Augustus De Morgan, Charles Sanders Peirce, and Ernst Schröder introduced several logical systems and calculi for binary relations which were explored in many ways in the following decades and became part of the emerging field of mathematical logic. Just as propositional logic is related to Boolean algebras, these calculi for binary relations also have an algebraic counterpart, called relation algebras[1]; in a nutshell, a relation algebra can be interpreted as a set of symbols for binary relations and the information of how these are interrelated. Formally speaking, a relation algebra is an algebra in the sense of universal algebra, i.e., a domain together with operations on it that satisfy certain axioms. The algebraic theory of relation algebras was first formulated by Alfred Tarski [Tar41]. With the triumph of propositional and predicate logics in the last 100 years, relational logics took more and more a back seat in modern mathematical logic. On the other hand, developments in computer science, particularly in fields such as database theory, automated reasoning and knowledge representation, have sparked a demand for robust formalisms to handle complex modeling and reasoning tasks. Since binary relations often play a prominent role in practical applications, it is not surprising that the old concept of relation algebras, with its origins in the early days of modern logic, has experienced a revival due to the rapid developments in computer science and Artificial Intelligence. Important examples of relation algebras that are interesting for applications are the Point Algebra, the Left Linear Point Algebra, Allen's Interval Algebra, and region connection calculi such as RCC5 and RCC8, just to name a few.

In this thesis we study a large class of computational decision problems that are induced in a natural way by relation algebras. These problems are called *network satisfaction problems* (*NSPs*). Consider a finite network in which the nodes are linked via the elements of a fixed relation algebra. Then the computational task of the NSP for this relation algebra is to decide whether the network satisfies a strong notion of consistency with respect to the fixed relation algebra. Such NSPs for a finite relation algebra can be used to model many computational problems in temporal and spatial reasoning [Dün05, RN07, BJ17]. Every

[1] Formal definitions of all terminology can be found in Chapter 2.

relation algebra gives rise to a specific network satisfaction problem whose computational complexity depends solely on the relation algebra itself. The central purpose of this work is to contribute to a better understanding of the complexities of these problems.

At this point, the reader might wonder how exactly the computational problems under consideration look like. We therefore present a simple example of a network satisfaction problem for a certain relation algebra, which is a good illustration of the kind of problems we are dealing with here. Consider a set of consecutive integer distances that contains 0, say $D = \{0, 1, 2, 3, 4, 5\}$. An instance of the computational problem is a network where each pair of nodes (x, y) has assigned a set of allowed distances $f(x, y) \subseteq D$. We can visualize an example of such an instance by the following edge-labeled graph:



The computational task now is to choose for each pair of nodes (x, y) a distance from $f(x, y)$ such that all triangle inequalities are satisfied in the whole network. For the instance above this is possible by the red marked distances. This computational decision problem is a typical example of a network satisfaction problem for a finite relation algebra. We want to mention that in this example the set of distances D plays the role of a relation algebra: we can think about D as a set of binary relations where our understanding of "distances" brings in certain assumptions about the behavior between elements of D, such as that there cannot be a triangle of points in the plane with distances 1, 1, and 3.[2]

There exist several different algorithms that can solve the network satisfaction problem for the fixed distance set $D = \{0, 1, 2, 3, 4, 5\}$. We could for example consider the brute force method that goes through all possible combinations of choices. For an input network with n nodes and $\binom{n}{2}$ edges, this algorithm requires about $5^{\binom{n}{2}}$ computation steps in the worst case to arrive at the correct decision. Since this is exponential in the input size n, the problem is in the complexity class EXPTIME[3]. Algorithms of exponential running time are often not useful in practice. Indeed, we can find a lower complexity class for our problem: since a guessed solution to the problem can be checked by an inspection of all triangles of the $\binom{n}{2}$ edges, the problem is in the complexity class NP. Such a non-deterministic polynomial-

[2] See Example 5.1.
[3] For an introduction to complexity theory see [AB09].

time algorithm cannot (probably) be implemented on a physical machine. An important question is whether there is a deterministic polynomial-time algorithm for solving our example problem; or in other words, does the problem fall into the complexity class P?[4]

It has probably not escaped the reader's attention that changing the set of distances for example to $D = \{0, \ldots, 13\}$ leads to a new decision problem and a new question about the complexity of that problem.[5] In fact, we can define a large class of decision problems in this way; for each finite initial segment $D \subset \mathbb{N}$ we get another problem and a priori a different answer to the question about the complexity.[6]

The question arises why there is a general interest in knowing the complexity of algorithms that solve computational problems? It is a simple but fundamental observation that today's world would be inconceivable without computers, or more precisely, without computation. The interest in knowledge about computational complexity can therefore be justified in several ways. Firstly, the complexity of concrete individual problems plays a fundamental role in the practical implementation and applicability of algorithms. For example, it is important to know how much time particular queries might take in a costumer database, navigation system, or an Internet search engine; computation is so ubiquitous that such a list can only be incomplete. Although computation or, more precisely, time-efficient computation is so pervasive, it is no exaggeration to say that our understanding of it is still in its infancy. By "understanding" we do not just mean knowing whether problems can be solved efficiently, but we mean a deeper comprehension of *why* these problems can be solved efficiently or why they cannot. This thesis is intended to contribute to a better understanding of why certain computational problems are efficiently solvable (and others probably not).

More than two decades ago, Robin Hirsch formulated in [Hir96] the "Really Big Complexity Problem (RBCP)":

RBCP: *Clearly map out which relation algebras are tractable and which are intractable.*

Hirsch asked for a classification of the computational complexity of the network satisfaction problem for every finite relation algebra. Such a classification should identify the NSPs that are solvable by a polynomial-time algorithm. For example, the complexity of the NSP for the earlier mentioned Point Algebra and Left Linear Point Algebra is in P [VKv90, BK07]. Both problems are used to model certain temporal scheduling tasks. However, the NSPs for Allen's Interval Algebra, RCC5, and RCC8 are NP-complete [All83, RN99]. These problems are well-known for their applications in spatial reasoning. There also exist relation algebras where the complexity of the network satisfaction problem is not in NP: Hirsch gave an example of a finite relation algebra with an undecidable network satisfaction problem [Hir99].

---

[4] The answer is yes.

[5] The problem is again in P.

[6] All those problems are in P, see Example 4.23.

All the mentioned complexity results are obtained for single examples of relation algebras and their NSPs, respectively. The same holds true in general for most of the complexity results in the field. Only a few results exist that are not a case-by-case study. It can be said that, measured against the size of Hirsch's RBCP, little has happened in the search for a general solution to it. Even 25 years later, we are still far from having a clear picture of the complexities of NSPs for finite relation algebras. Whenever a classification task seems too big for the methods and tools one has at hand, it may be advisable to tackle the classification on a suitable subclass first. The first important restriction to a subclass of NSPs was made by Hirsch himself, when he introduced relation algebras with *normal representations* [Hir96].

Relation Algebras with a Normal Representation

We want to give an idea of normal representations in order to be able to formulate the results of this work, but we should point out that the intuition given here is not intended to replace the formal definitions given in Chapter 2. In the following, we will denote a relation algebra by $\mathbf{A}$ and its network satisfaction problem by $\mathrm{NSP}(\mathbf{A})$. For the moment, it is sufficient to think of $\mathbf{A}$ as a set of symbols for binary relations A together with a set of rules describing how they "compose" and whether they "contain" each other. A representation $\mathfrak{B}$ is a relational structure that has only binary relations and the set of these relations contains at least the empty relation, the full relation, and equality and is closed under intersection, union, complement, converse, and composition of relations[7]. By definition, the set of relations of a representation $\mathfrak{B}$ naturally carries a Boolean algebra with set-wise union, intersection, and complement as operations. The order associated with this Boolean algebra is simply the set-wise inclusion order. Consider the relational structure $(\mathbb{Q};\ \emptyset, =, <, >, \leqslant, \geqslant, \neq, \mathbb{Q}^2)$ as an example. The domain of the structure is the set of rational numbers $\mathbb{Q}$ and the binary relations $\{\emptyset, =, <, >, \leqslant, \geqslant, \neq, \mathbb{Q}^2\}$ are meant to be the usual ones on $\mathbb{Q}$. It is easy to see that this structure is a representation, since the set of relations contains the empty relation, the full relation, and equality and is closed under intersection, union, complement, converse, and composition of relations. For example, it holds that $< \circ <$ is equal to $<$. As mentioned before, the order induced by the Boolean algebra is the set-wise inclusion order $\subset$; we have for example $< \subset \leqslant$.

We say that $\mathfrak{B}$ is a representation of the relation algebra $\mathbf{A}$ if $\mathfrak{B}$ is a representation whose relations correspond exactly to the symbols of $\mathbf{A}$ in a way that respects the rules in $\mathbf{A}$. That is, $\mathfrak{B}$ contains a concrete binary relation for each symbol of A, and these concrete relations behave exactly as described by the composition and containment rules in $\mathbf{A}$. In general, neither the existence nor the uniqueness of representations for a relation algebra is guaranteed.

A normal representation $\mathfrak{B}$ is a certain kind of representation where we have stronger requirements: Firstly, we require that every two elements of $\mathfrak{B}$ are in some binary relation

[7]The composition $R \circ D$ of two binary relations is defined as $R \circ D := \{(x,z) \mid \exists y.((x,y) \in R \wedge (y,z) \in D)\}$.

from the signature of $\mathfrak{B}$. Secondly, the representation should be universal for all representations of $\mathbf{A}$, i.e., $\mathfrak{B}$ is in a sense the most general representation of $\mathfrak{B}$. And last but not least, we want $\mathfrak{B}$ to satisfy a strong model theoretical property: every isomorphism between finite substructures of $\mathfrak{B}$ can be extended to an automorphism of the whole structure $\mathfrak{B}$. This property is known as *homogeneity*. The study of homogeneous structures has a long history in the areas of model theory, permutation group theory, and combinatorics.[8] Two well-known examples of homogeneous structures are the ordered rationals $(\mathbb{Q};<)$ and the countable universal homogeneous graph[9].

From today's perspective, focusing on relation algebras with normal representations is so promising because their NSPs are closely related to a different class of computational problems for which a rich theory has been developed in the last decades. We are talking about *constraint satisfaction problems over homogeneous domains*.

A *constraint satisfaction problem (CSP)* can be defined as the computational problem of deciding whether a given finite relational structure has a homomorphism to a fixed structure of the same signature. In Chapter 2 we present a translation of network satisfaction problems to constraint satisfaction problems. Most of our findings are based on the application of the *universal algebraic approach*, a powerful theory for studying CSPs. We do not want to go into details here, but refer to Section 2.4 and Section 2.6.2, which serve as introductions to CSPs and the universal algebraic approach. Instead, we take a look at the results and structure of this thesis.

Contributions and Organization of the Thesis

The interested reader will find in Chapter 2 important terminology and detailed basic knowledge, especially about relation algebras, CSPs, and universal algebra, which is indispensable for reading the thesis. The results of this work are divided into three parts as follows.

Chapter 3: Hardness Criteria for NSPs

We provide in this chapter two general conditions that imply NP-hardness of network satisfaction problems for finite relation algebras. Stated formally we prove the following:

Theorem 1 *Let* $\mathbf{A}$ *be a finite relation algebra with a normal representation* $\mathfrak{B}$. *Assume that one of the following holds:*

1. $\mathfrak{B}$ *contains a non-trivial equivalence relation with a finite number of equivalence classes.*

2. $\mathfrak{B}$ *has domain size at least three and contains no non-trivial equivalence relation, but contains a relation* $a^{\mathfrak{B}}$ *such that*

[8] See the survey [Mac11].

[9] Also known as the Rado graph or Random graph.

- $a^{\mathfrak{B}}$ *is symmetric, i.e.,* $(x, y) \in a^{\mathfrak{B}}$ *if and only if* $(y, x) \in a^{\mathfrak{B}}$*,*
- *for the binary relational product* $\circ$ *it holds that* $a^{\mathfrak{B}} \nsubseteq a^{\mathfrak{B}} \circ a^{\mathfrak{B}}$*.*

Then $\mathrm{NSP}(\mathbf{A})$ *is NP-hard.*

We use this result to complete the classification of the so called small relation algebras by Hirsch and Cristiani [CH04]. An overview of the methods we use to establish these results can be found in the introduction to Chapter 3.

This part mainly contains published results from the article

[BK20] Manuel Bodirsky and Simon Knäuer, *Hardness of network satisfaction for relation algebras with normal representations*, Relational and Algebraic Methods in Computer Science, Springer International Publishing, 2020, pp. 31–46.

Chapter 4: Symmetric Relation Algebras with a Flexible Atom

In this part we completely classify the computational complexities of NSPs for finite *symmetric* relation algebras $\mathbf{A}$ with a *flexible atom*. A relation algebra is called symmetric if all relations of any of its representations $\mathfrak{B}$ are symmetric, i.e., $(x, y) \in a^{\mathfrak{B}}$ if and only if $(y, x) \in a^{\mathfrak{B}}$ holds for all relations $a^{\mathfrak{B}}$ from $\mathfrak{B}$. A flexible atom is a relation $s^{\mathfrak{B}}$ that is contained in every non-trivial relational product [Com84, Mad82]. Formally speaking, $s^{\mathfrak{B}} \subseteq a^{\mathfrak{B}} \circ b^{\mathfrak{B}}$ holds for all non-reflexive relations $a^{\mathfrak{B}}$ and $b^{\mathfrak{B}}$ from $\mathfrak{B}$. The existence of a flexible atom is a simple condition which implies that a finite relation algebra $\mathbf{A}$ has a normal representation and hence we can deal with its NSP by means of the algebraic methods mentioned earlier. We want to remark that such relation algebras have been studied intensively, for example in the context of the so-called *flexible atoms conjecture* [Mad94, AMM08].

It will be useful in the following to denote by A_0 the subset of the domain of $\mathbf{A}$ that consists of *atoms*. Atoms are elements that correspond to set-wise minimal relations in representations. Furthermore, we will talk about the *allowed triples* of a relation algebra $\mathbf{A}$, by which we mean triples (a, b, c) of elements from A_0 such that there exists a representation $\mathfrak{B}$ of $\mathbf{A}$ with $c^{\mathfrak{B}} \subseteq a^{\mathfrak{B}} \circ b^{\mathfrak{B}}$.

Our classification result states that every NSP of a finite symmetric relation algebra with a flexible atom is either solvable in polynomial-time or is NP-complete, unless $\mathrm{P} = \mathrm{NP}$. Note that this is already a highly non-trivial result, since a famous classical result of Ladner proves the existence of NP-intermediate problems, i.e. problems in NP that are neither in P nor NP-complete (unless $\mathrm{P} = \mathrm{NP}$). However, our result even provides a criterion for the NSP of a finite relation algebra to be in P. We prove the following:

Theorem 2 *Let* $\mathbf{A}$ *be a finite symmetric relation algebra with a flexible atom, and let* A_0 *be the set of atoms of* $\mathbf{A}$*. Then one of the following holds:*

- *There exists an operation* $f\colon A_0^6 \to A_0$ *such that*
  1. f *preserves the allowed triples of* $\mathbf{A}$,
  2. $\forall x_1, \dots, x_6 \in A_0.\, f(x_1, \dots, x_6) \in \{x_1, \dots x_6\}$,
  3. f *satisfies the Siggers identity:* $\forall x, y, z \in A_0.\, f(x, x, y, y, z, z) = f(y, z, x, z, x, y)$,

  in this case the network satisfaction problem for $\mathbf{A}$ *is in P.*
- *The network satisfaction problem for* $\mathbf{A}$ *is NP-complete.*

It is easy to see that for a finite relation algebra $\mathbf{A}$, checking the existence of an operation satisfying the first condition is a decidable task and can be done by a computer. This means that we can use the criterion to determine whether the NSP is in P for a concretely given relation algebra. With this result we solve Hirsch's RBCP for finite symmetric relation algebras with a flexible atom.

The content of Chapter 4 consists largely of the article

[BK21] Manuel Bodirsky and Simon Knäuer, *Network satisfaction for symmetric relation algebras with a flexible atom*, Proceedings of the AAAI Conference on Artificial Intelligence **35** (2021), no. 7, 6218–6226.

An extension of this article has been submitted for publication in a peer-reviewed journal.

Chapter 5: A Datalog versus NP-complete Dichotomy

In Chapter 5 we investigate a class of NSPs which are solved by *Datalog programs*. Datalog is a fundamental formalism in theoretical computer science, which is intensively studied, e.g. in database theory. The class of algorithms and procedures we want to define in this work via Datalog programs has many other names in the literature. The idea is always to establish some form of local consistency. Consider for example the "distance problem" from page 2. The question is whether it is possible to make the right decision for a given distance-network while inspecting only a fixed number of nodes (e.g., the three nodes of a triangle) at a time.[10] Since Datalog programs can be executed on finite structures in polynomial time, the class of computational problems that can be decided by a Datalog program is a subclass of P. In the study of NSPs, it is generally assumed that consistency algorithms solve most polynomial-time solvable problems, although there are examples of NSPs where this is not the case.[11] Our result strongly supports this assumption. We prove the following.

Theorem 3 *Let* $\mathbf{A}$ *be a finite symmetric relation algebra with a normal representation* $\mathfrak{B}$ *that has an injective polymorphism, and let* A_0 *be the set of atoms of* $\mathbf{A}$. *If there exists an operation* $f\colon A_0^6 \to A_0$ *such that*

[10] All those problems can be solved by Datalog programs, see Example 5.1.
[11] See Example 5.22.

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

then the network satisfaction problem for $\mathbf{A}$ can be solved by a Datalog program.

We can apply this result to the class of relation algebras from the previous chapter and show that every polynomial-time solvable NSP of a symmetric relation algebra with a flexible atom can even be solved by a Datalog program (unless P = NP). In combination with Theorem 2 this proves a Datalog versus NP-complete dichotomy for NSPs of finite symmetric relation algebras with a flexible atom.

We obtain these results by analyzing the algebraic properties of a certain class of finite structures. The results in Chapter 5 are not yet published.

The author obtained further results during his doctoral studies [BKS20, BKR21, BFKR21]. These results are not part of the thesis which focuses on network satisfaction problems of relation algebras.

Chapter 2

Background

2.1 Structures and Morphisms

We start this chapter with an overview of the basics of structures and mappings between them. Let $\mathbb{N}$ denote the natural numbers starting with 0 and define $\mathbb{N}_+ := \mathbb{N}\backslash\{0\}$. A *signature* τ is a set of relation symbols and function symbols. Each symbol is associated with a natural number, called the *arity* of the symbol. Function symbols of arity 0 are called *constant symbols*. A τ-structure is a tuple $\mathfrak{A} = (A; (Q^{\mathfrak{A}})_{Q\in\tau})$ where A is a set, called the *domain* of $\mathfrak{A}$, such that for every $Q \in \tau$:

- if Q is a relation symbol of arity $n \in \mathbb{N}$, then $Q^{\mathfrak{A}}$ is a subset of A^n,
- if Q is a function symbol of arity $n \in \mathbb{N}$, then $Q^{\mathfrak{A}}$ is an operation $A^n \to A$.

Note that by $A^0 = \{\varnothing\}$ a subset of A^0 can be seen as a Boolean value and the operation $f\colon A^0 \to A$ can be interpreted as a constant. As long as there is no risk of confusion we will often use the function symbols for the corresponding functions, and the relation symbols for the corresponding relations, i.e., we use Q instead of $Q^{\mathfrak{A}}$. A τ-structure is called *finite* if its domain is finite.

Let $\mathfrak{A}$ and $\mathfrak{B}$ be two τ-structures. A *homomorphism* h from $\mathfrak{A}$ to $\mathfrak{B}$ is a function $h\colon A \to B$ such that

- for every relation symbol Q of arity $n \in \mathbb{N}$ and every tuple $(a_1, \ldots, a_n) \in A^n$, we have that $(a_1, \ldots, a_n) \in Q^{\mathfrak{A}} \Rightarrow (h(a_1), \ldots, h(a_n)) \in Q^{\mathfrak{B}}$;
- for every function symbol Q of arity $n \in \mathbb{N}$ and every tuple $(a_1, \ldots, a_n) \in A^n$, we have that $h(Q^{\mathfrak{A}}(a_1, \ldots, a_n)) = Q^{\mathfrak{B}}(h(a_1), \ldots, h(a_n))$.

In the case that τ contains only function symbols and h is surjective, then $\mathfrak{B}$ is called *homomorphic image* of $\mathfrak{A}$. In general, the homomorphism h is called an *embedding* if h is injective and satisfies

- for every relation symbol Q of arity $n \in \mathbb{N}$ and every tuple $(a_1, \ldots, a_n) \in A^n$, we have that $(a_1, \ldots, a_n) \in Q^{\mathfrak{A}} \Leftrightarrow (h(a_1), \ldots, h(a_n)) \in Q^{\mathfrak{B}}$.

A surjective embedding is called an *isomorphism*. An *endomorphism* of a τ-structure $\mathfrak{A}$ is a homomorphism from $\mathfrak{A}$ to $\mathfrak{A}$ and an *automorphism* of $\mathfrak{A}$ is an isomorphism from $\mathfrak{A}$ to $\mathfrak{A}$. We denote by $\mathrm{Aut}(\mathfrak{A})$ the group of all *automorphisms* of $\mathfrak{A}$. For every k-tuple $(a_1, \ldots, a_k) \in A^k$ we call the set $L = \{(\alpha(a_1), \ldots, \alpha(a_k)) \mid \alpha \in \mathrm{Aut}(\mathfrak{A})\}$ a *k-orbit of* $\mathrm{Aut}(\mathfrak{A})$.

A τ-structure $\mathfrak{A}$ is a *substructure of a τ-structure* $\mathfrak{B}$ if

- $A \subseteq B$;
- for every relation symbol Q of arity $n \in \mathbb{N}$ and every tuple $(a_1, \ldots, a_n) \in A^n$, we have that $(a_1, \ldots, a_n) \in Q^{\mathfrak{A}}$ if and only if $(a_1, \ldots, a_n) \in Q^{\mathfrak{B}}$;
- for every function symbol Q of arity $n \in \mathbb{N}$ and every tuple $(a_1, \ldots, a_n) \in A^n$, we have that $Q^{\mathfrak{A}}(a_1, \ldots, a_n) = Q^{\mathfrak{B}}(a_1, \ldots, a_n)$.

Note that for every subset $A' \subseteq B$ of the domain of a τ-structure $\mathfrak{B}$ there exists a unique substructure $\mathfrak{A}$ of $\mathfrak{B}$ with smallest domain A and $A' \subseteq A$. We call this the *substructure of* $\mathfrak{B}$ *induced by* A. A relational structure is called *connected* if it is not the disjoint union of two structures. A *connected component* of a relational structure $\mathfrak{B}$ is a with respect to domain inclusion maximal substructure $\mathfrak{A}$ of $\mathfrak{B}$ that is connected.

Let τ and ρ be relational signatures such that $\tau \subseteq \rho$ and let $\mathfrak{A}$ be a τ-structure and $\mathfrak{A}'$ be a ρ-structure with the same domain. If $R^{\mathfrak{A}} = R^{\mathfrak{A}'}$ holds for every $R \in \tau$ we say that $\mathfrak{A}$ is the *τ-reduct* of $\mathfrak{A}'$ and $\mathfrak{A}'$ is a *ρ-expansion* 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 $((a_1, b_1), \ldots, (a_n, b_n)) \in (A \times B)^n$, we have that $((a_1, b_1), \ldots, (a_n, b_n)) \in Q^{\mathfrak{C}}$ if and only if $(a_1, \ldots, a_n) \in Q^{\mathfrak{A}}$ and $(b_1, \ldots, b_n) \in Q^{\mathfrak{B}}$;
- for every function symbol Q of arity $n \in \mathbb{N}$ and every tuple $((a_1, b_1), \ldots, (a_n, b_n)) \in (A \times B)^n$, we have that

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

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 structure with purely relational signature are called *relational structures*. Since we do not deal with signatures of mixed type in this thesis, we will from now on use the term structure for relational structures only.

We will later need the following notions for algebras.

2.1 Definition Let $\mathcal{K}$ be a class of algebras. Then we have

- $\mathrm{H}(\mathcal{K})$ is the class of homomorphic images of algebras from $\mathcal{K}$ and
- $\mathrm{S}(\mathcal{K})$ is the class of subalgebras of algebras from $\mathcal{K}$.
- $\mathrm{P}^{\mathrm{fin}}(\mathcal{K})$ is the class of finite products of algebras from $\mathcal{K}$.

We close this section with a first definition of constraint satisfaction problems. Note that we introduce this computational decision problem using only the terms of this section.

2.2 Definition Let $\mathfrak{B}$ be a relational τ-structure. The *constraint satisfaction problem* of $\mathfrak{B}$ is a computational decision problem of the following form:
Input: A finite τ-structure $\mathfrak{A}$.
Question: Is there a homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$?

This computational decision problem is denoted by $\mathrm{CSP}(\mathfrak{B})$. The structure $\mathfrak{B}$ is called the *template* or the *domain* of $\mathrm{CSP}(\mathfrak{B})$.

2.2 First-Order Logic

We assume basic knowledge in classical first-order logic and recommend the textbook [Hod97] for a detailed introduction to the topic. We recall in this paragraph some notions that play a role in this thesis.

We make use of first-order logic that allows the formulas $x = y$ (for equality), $\bot$ (for 'false'), and $\top$ (for 'true') for every signature τ, unless stated otherwise. A first-order formula without free variables is called a (*first-order*) *sentence*. A formula without existential and universal quantifiers is called *quantifier-free*.

Let τ be a relational signature. A first-order τ-formula $\varphi(x_1, \ldots, x_n)$ is called *primitive positive* (*pp*) if it has the form

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

where $\varphi_1, \ldots, \varphi_s$ are atomic τ-formulas, i.e., formulas of the form $R(y_1, \ldots, y_l)$ for $R \in \tau$ and $y_i \in \{x_1, \ldots, x_m\}$, of the form $y = y'$ for $y, y' \in \{x_1, \ldots x_m\}$, or of the form $\bot$.

Let $\mathfrak{A}$ be a τ-structure and $\varphi(x_1, \ldots, x_n)$ be a first-order τ-formula. A relation $R \subseteq A^n$ is *definable over* $\mathfrak{A}$ *by* $\varphi(x_1, \ldots, x_n)$ if

$$R = \{(a_1, \ldots, a_n) \in A^n \mid \mathfrak{A} \models \varphi(a_1, \ldots, a_n)\}.$$

We say that a relation R is *first-order* (*or primitively positively, or quantifier-free*) *definable over* $\mathfrak{A}$ if there exists a first-order (or primitive positive, or quantifier-free) τ-formula $\varphi(x_1, \ldots, x_n)$ such that R is definable over $\mathfrak{A}$ by $\varphi(x_1, \ldots, x_n)$. The set of relations that are first-order definable over a τ-structure $\mathfrak{A}$ is denoted by $\langle \mathfrak{A} \rangle_{fo}$ and analogously the primitively positively definable relations are $\langle \mathfrak{A} \rangle_{pp}$.

The notion of primitive positive formulas is central for this thesis. The reason for this is the following computational problem which is up to a logspsace reduction equivalent to CSP($\mathfrak{B}$).

2.3 Definition Let $\mathfrak{B}$ be a relational τ-structure. Then $\mathrm{CSP}^{pp}(\mathfrak{B})$ is the computational decision problem:

Input: A primitive positive sentence φ.

Question: Does φ hold in $\mathfrak{B}$?

We will see in Section 2.4 the details of the translation between CSP($\mathfrak{B}$) and $\mathrm{CSP}^{pp}(\mathfrak{B})$.

2.3 Relation Algebras

In this section we introduce the basic definitions and properties of relation algebras. We start with the intuitive notion of proper relation algebras, move on to abstract relation algebras and explain finally what representations of relation algebras are. For an introduction into the field we recommend the textbook by Maddux [Mad06b]. We use bold letters (such as **A**) to denote relation algebras of all kinds and the corresponding roman letter (such as A) to denote the domain of the algebra.

2.3.1 Proper Relation Algebras

Proper relation algebras are algebras that have a set of binary relations as their domain and are closed under certain operations.

2.4 Definition Let D be a set and $E \subseteq D^2$ an equivalence relation on D and $\mathcal{P}(E)$ the power set of E. Let $(\mathcal{P}(E);\ \cup,\ ^-,\ 0,\ 1,\ \mathrm{Id},\ ^\smile,\ \circ)$ be an algebra with the following operations:

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

for $a, b \in \mathcal{P}(E)$. A subalgebra of $(\mathcal{P}(E);\ \cup,\ ^-,\ 0,\ 1,\ \mathrm{Id},\ ^\smile,\ \circ)$ is called a *proper relation algebra*.

The class of all proper relation algebras is denoted by PA. 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 a *representable relation algebra*. We denote the class of representable relation algebras by RRA.

2.5 Example Consider the set $\mathbb{Q}$ of rational numbers. The set $\mathbb{Q}^2$ is clearly an equivalence relation and therefore the algebra $\mathbf{A} = (\mathcal{P}(\mathbb{Q}^2);\ \cup,\ ^-,\ 0,\ 1,\ \mathrm{Id},\ ^\smile,\ \circ)$ is a proper relation algebra. Furthermore, one can check that the set $P = \{\emptyset, <, >, =, \leqslant, \geqslant, \neq, \mathbb{Q}^2\} \subset \mathcal{P}(\mathbb{Q}^2)$ induces a subalgebra **P** of **A** with domain P (i.e., the set P is closed under the application of the operations from **A**). The proper relation algebra **P** is called *point algebra*.

2.3.2 Abstract Relation Algebras

We introduce abstract relation algebras as algebras with the signature $\{\cup, \overline{}, 0, 1, \mathrm{Id}, \smile, \circ\}$ that satisfy certain identities. It is not common to use the same signature for abstract relation algebras and proper relation algebras. Nevertheless, we have decided to do so in the interest of better readability. However, we would like to emphasize at this point that depending on the context, the symbol $\cup$ for example could have the following different meanings:

1. just a function symbol of arity 2, e.g., as an element of a signature,
2. the 2-ary set union operation, e.g., in a proper relation algebra,
3. an arbitrary 2-ary operation on elements that are not necessary relations or sets, e.g., as in an abstract relation algebra.

It is always clear from the context which meaning is intended. Especially for representable relation algebras, which is the kind of relation algebra we are mainly interested in (see Definition 2.13 for the reason) it will be natural to switch between 2. and 3. above; the definition of a representation of a relation algebra will build a bridge between abstract and proper relation algebras.

2.6 Definition An (*abstract*) *relation algebra* **A** is an algebra with domain A and signature $\{\cup, \overline{}, 0, 1, \mathrm{Id}, \smile, \circ\}$ such that

1. the structure $(A; \cup, \cap, \overline{}, 0, 1)$, with $\cap$ defined by $x \cap y := \overline{(\overline{x} \cup \overline{y})}$, 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 $a \in A$: $(a^{\smile})^{\smile} = a$,
6. for all $a, b \in A$: $(a \cup b)^{\smile} = a^{\smile} \cup b^{\smile}$,
7. for all $a, b \in A$: $(a \circ b)^{\smile} = b^{\smile} \circ a^{\smile}$,
8. for all $a, b, c \in A$: $\overline{b} \cup (a^{\smile} \circ \overline{((a \circ b)}) = \overline{b}$.

We denote the class of all abstract relation algebras by RA. Since every proper relation algebra and therefore also every representable relation algebra satisfies the axioms from the previous definition we have PA $\subset$ RRA $\subset$ RA. It is a classical result by [Lyn50] that there exist finite relation algebras $\mathbf{A} \in$ RA that are not representable relation algebras; so the inclusions above are indeed proper inclusions.

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

Figure 2.1: Multiplication table of the point algebra **P**.

Let $\mathbf{A} = (A;\ \cup,\ ^{-},\ 0,\ 1,\ \mathrm{Id},\ ^{\smile},\ \circ)$ be a relation algebra. We define the new operation $x \cap y := \overline{\overline{x} \cup \overline{y}}$ on the set A. Then the algebra $(A;\ \cup,\ \cap,\ ^{-},\ 0,\ 1)$ is by definition a Boolean algebra and induces therefore a partial order $\leqslant$ on A, which is defined by $x \leqslant y :\Leftrightarrow x \cup y = y$. Note that for proper relation algebras, this ordering coincides with the set-inclusion order. The minimal elements of this order in $A \backslash \{0\}$ are called *atoms*. The set of atoms of **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 **A** with subsets of A_0. Especially when defining concrete relation algebras, we will often only name the atoms explicitly and define all other elements as subsets of atoms. For more information about Boolean algebras, we recommend the book [HG09].

By item 3) in Definition 2.6 the values of the composition operation $\circ$ in **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. We illustrate this with the following example.

2.7 Example Recall the definition of the point algebra **P** from Example 2.5. The set of atoms of **P** is $P_0 = \{\mathrm{Id}, <, >\}$. By the definition of the composition operation $\circ$ in the proper relation algebra **P** we get the multiplication table in Figure 2.1 for the values of the composition $\circ$ on P_0.

We call **A** *symmetric* if all its elements are symmetric, i.e., $a^{\smile} = a$ for every $a \in A$. The point algebra is clearly an example of a relation algebra that is not symmetric.

2.3.3 Representations

By the previous definition a relation algebra $\mathbf{A} \in \mathrm{RRA}$ is a representable relation algebra if it has an isomorphism to a proper relation algebra. Such an isomorphism is usually called the representation of **A**. Since we will be interested in the model-theoretic behavior of a set of relations which is the domain of a proper relation algebra, we consider a relational structure that "realizes" the proper relation algebra with its relations. It will be convenient for us to call this relational structure a representation and to not use the classical notion here. However, it is easy to see that the existence of a representation is equivalent under both definitions.

2.8 Definition Let $\mathbf{A} \in \mathrm{RRA}$. Then 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 $\mathbf{A}$ and the proper relation algebra induced by the relations of $\mathfrak{B}$ in $(\mathcal{P}(1^{\mathfrak{B}});\ \cup,\ ^{-},\ 0,\ 1,\ \mathrm{Id},\ ^{\smile},\ \circ)$.

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 thesis.

2.9 Definition A tuple $(x, y, z) \in (A_0)^3$ is called an *allowed triple* (*of* $\mathbf{A}$) if $z \leqslant x \circ y$. Otherwise, (x, y, z) is called a *forbidden triple* (*of* $\mathbf{A}$) and 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 exists $b_1, b_2, b_3 \in B$ 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.

2.10 Example The structure $(\mathbb{Q};\ \emptyset, <, >, =, \leqslant, \geqslant, \neq, \mathbb{Q}^2)$ is a representation of the point algebra $\mathbf{P}$ from Example 2.5 and Example 2.7. The triple $(<, <, <)$ is an example of an allowed triple of $\mathbf{P}$ and the triple $(<, <, =)$ is an example of a forbidden triple.

A relation $R \subseteq A^3$ is called *totally symmetric* (cf. Definition 4.19) if for all bijections $\pi\colon \{1, 2, 3\} \to \{1, 2, 3\}$ 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.

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

2.3.4 Network Satisfaction Problem

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

2.12 Definition 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 $\mathfrak{B}$.

$\circ$	Id	a	b
Id	Id	a	b
a	a	$\mathrm{Id} \cup b$	$a \cup b$
b	b	$a \cup b$	$\mathrm{Id} \cup a \cup b$

Figure 2.2: Multiplication table of the relation algebra #17.

With these notions we can define the network satisfaction problem.

2.13 Definition 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.

Consider the following example of the network satisfaction problem for a concrete relation algebra. The numbering of the relation algebra is by [AM94].

2.14 Example (An instance of NSP of relation algebra #17) Let $\mathbf{A}$ be the representable relation algebra with the set of atoms $\{\mathrm{Id}, a, b\}$ and the values for the composition operation $\circ$ on these atoms be given by Table 2.2. Note that this determines the composition operation on the whole domain of $\mathbf{A}$, which is the following set:

$$A = \{\varnothing, \mathrm{Id}, a, b, \mathrm{Id} \cup a, \mathrm{Id} \cup b, a \cup b, \mathrm{Id} \cup a \cup b\}.$$

Let $V := \{x_1, x_2, x_3\}$ be a set. Consider the map $f \colon V^2 \to A$ given by

$$\begin{aligned} &f(x_i, x_i) = \mathrm{Id} \quad \text{for all } i \in \{1,2,3\} \\ &f(x_1, x_2) = f(x_2, x_1) = a \\ &f(x_1, x_3) = f(x_3, x_1) = \mathrm{Id} \cup a \\ &f(x_2, x_3) = f(x_3, x_2) = b \cup a. \end{aligned}$$

The tuple $(V; f)$ is an example of an instance of NSP of $\mathbf{A} \in \mathrm{RRA}$.

We will in the following assume that for an $\mathbf{A}$-network $(V; f)$ it holds that $f(V^2) \subseteq A \backslash \{0\}$. 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 $f(x, y) = a$.

2.3.5 Normal Representations

In this section we consider a subclass of RRA introduced by Hirsch in 1996. For relation algebras $\mathbf{A}$ from this class, $\mathrm{NSP}(\mathbf{A})$ corresponds naturally to a constraint satisfaction problem.

In the following let $\mathbf{A}$ be in RRA. An $\mathbf{A}$-network $(V; f)$ is called *closed* (transitively closed in the work by Hirsch [Hir97]) if f is total and for all $x, y, z \in V$ it holds that $f(x,x) \leqslant \mathrm{Id}$, $f(x,y) = f(y,x)^{\smile}$, and $f(x,z) \leqslant f(x,y) \circ f(y,z)$. It is called *atomic* if the range of f only contains atoms from $\mathbf{A}$.

2.15 Definition (from [Hir96]) 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 will now investigate the connection between $\mathrm{NSP}(\mathbf{A})$ for a finite relation algebra with a normal representation $\mathfrak{B}$ and constraint satisfaction problems. Recall the definition of $\mathrm{CSP}^{pp}(\mathfrak{B})$ for a relational structure $\mathfrak{B}$ which involves primitive positive sentences (Definition 2.3). Consider the following translation which associates to each $\mathbf{A}$-network $(V; f)$ a primitive positive A-sentences φ as follows: the variables of φ are the elements of V and φ contains for every (x,y) in the domain of f the conjunct $a(x,y)$ if and only if $f(x,y) = a$ holds. For the other direction let φ be an A-sentence with variable set X and consider the $\mathbf{A}$-network $(X; f)$ with the following definition: for every $x, y \in X$, if (x,y) does not appear in any conjunct from φ we leave $f(x,y)$ undefined, otherwise let $a_1(x,y), \ldots, a_n(x,y)$ be all the conjuncts from φ that contain (x,y). We compute in $\mathbf{A}$ the element $a := a_1 \cap \ldots \cap a_n$ and define $f(x,y) := a$.

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

2.16 Theorem (Proposition 1.3.16 in [Bod12], see also [BJ17, Bod18]) *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* $\mathrm{CSP}^{pp}(\mathfrak{B})$ *are the same problem up to the translation between* $\mathbf{A}$*-networks and* A*-sentences.*
2. *If* $\mathbf{A}$ *has a normal representation* $\mathfrak{B}$ *the problems* $\mathrm{NSP}(\mathbf{A})$ *and* $\mathrm{CSP}^{pp}(\mathfrak{B})$ *are the same up to the translation between* $\mathbf{A}$*-networks and* A*-sentences.*

Before we take a closer look at this translation and its consequences, we must finally get to know constraint satisfaction problems better.

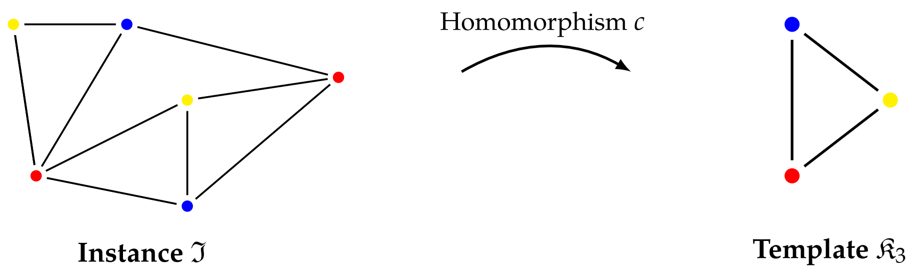


Figure 2.3: A satisfiable instance of the 3-coloring problem from Example 2.17.

2.4 Constraint Satisfaction Problems

Constraint satisfaction problems form a large class of computational decision problems. Many classical decision problems such as the satisfiability problem for propositional formulas, the satisfiability problem for systems of linear equations over a (finite) field or the graph coloring problems. To get an idea of the problems, let us take a closer look at an example of a graph coloring problem.

2.17 Example Consider the undirected complete graph on 3 vertices $\{b, r, y\}$. This relational structure is denoted by $\mathfrak{K}_3$. We now take an arbitrary relational structure $\mathfrak{I}$ in the binary signature of graphs. According to Definition 2.2 the structure $\mathfrak{I}$ is accepted as an instance of $\mathrm{CSP}(\mathfrak{K}_3)$ if there exists a homomorphism from $\mathfrak{I}$ to $\mathfrak{K}_3$. If we interpret the domain elements of $\mathfrak{K}_3$ as three different colors, this means that there is a vertex coloring of $\mathfrak{I}$, so that two adjacent vertices get different colors; see Figure 2.3. It turns out that $\mathrm{CSP}(\mathfrak{K}_3)$ is exactly the 3-coloring problem for graphs. More general for every $n \in \mathbb{N}_+$ the problem $\mathrm{CSP}(\mathfrak{K}_n)$ is the n-coloring problem for graphs.

The reader may recall that we have already seen two definitions of constraint satisfaction problems, the definition of $\mathrm{CSP}(\mathfrak{B})$ in Section 2.1 and the definition of $\mathrm{CSP}^{pp}(\mathfrak{B})$ in Section 2.2. The following folklore result justifies to switch between the two definitions.

2.18 Proposition *Let τ be a finite relational signature and let $\mathfrak{B}$ be a τ-structure. Then* $\mathrm{CSP}(\mathfrak{B})$ *and* $\mathrm{CSP}^{pp}(\mathfrak{B})$ *are equivalent up to logspace reductions.*

Proof: For every primitive positive τ-sentence φ with variable set X the *canonical database* $D(\varphi)$ is defined as the τ-structure on X where $x \in X^m$ is in the relation $R^{D(\varphi)}$ for $R \in \tau$ if and only if $R(x)$ is a conjunct from φ. Conversely every relational τ-structure $\mathfrak{A}$ induces a primitive positive τ-sentence on the variable set A, the so-called *conjunctive query*, simply by adding to the conjunction all atomic τ-formulas that hold in $\mathfrak{A}$. It is easy to see that the canonical database and the conjunctive query can be computed in logspace and preserve the acceptance condition of the two computational problems. □

At the beginning of this section we made the informal statement that the class of CSPs is very large. The following theorem provides some evidence for this.

2.19 Theorem (Theorem 1 in [BG08]) *Every computational decision problem is polynomial-time Turing equivalent to* $\mathrm{CSP}(\mathfrak{B})$ *for some structure* $\mathfrak{B}$.

However there are several interesting subclasses of the class of all CSPs where the complexities of the problems fall into extensively studied complexity classes, such as the prominent classes P and NP.[1] One of the most iridescent problems in computer science and mathematics of the last 50 years is the question whether $\mathrm{P} = \mathrm{NP}$ holds. This question is wide open. Simply speaking, we do not know whether the problems whose solutions can be efficiently checked, can also be efficiently solved, i.e. whether $\mathrm{NP} \subseteq \mathrm{P}$.

It is known from a famous result by Ladner that $\mathrm{P} \neq \mathrm{NP}$ would imply the existence of so-called *NP-intermediate* problems [Lad75]; under the assumption that $\mathrm{P} \neq \mathrm{NP}$ one can construct problems that are neither in P nor NP-complete. Ladner's "diagonalization construction" gives rise to very artificial NP-intermediate problems and today it is still an interesting question whether there might exist "more natural" NP-intermediate problems. One way to tackle this question is to study classes of computational problems which are widely thought to be "natural" and then answer the question whether there exist NP-intermediate problems within this class or not (unless $\mathrm{P} = \mathrm{NP}$). The class of CSPs with finite templates is an example of such a "natural" class of problems. It is easy to see that all these finite-domain CSPs are in NP; a guessed solution (i.e., a homomorphism or a variable assignment) can be checked in polynomial-time. Feder and Vardi [FV99] initiated in their pioneering work a systematic view on the complexities of those CSPs. They conjectured that the class of finite-domain CSPs, despite its richness admits a complexity dichotomy:

2.20 Conjecture (Feder-Vardi Dichotomy Conjecture) *Every finite-domain CSP is either in P or NP-complete, unless* $\mathrm{P} = \mathrm{NP}$.

The conjecture and the search for a positive answer led to a new theory which combines universal algebra and complexity theory (see Section 2.6 for an introduction to the details). This so-called *universal algebraic approach* even resulted in proofs of the Feder-Vardi conjecture: after being unsolved for two decades, Bulatov and Zhuk independently proved its correctness in 2017. In the last 20 years, a number of algebraic criteria have been found that imply the NP-hardness of a CSP. Bulatov and Zhuk then finally found algorithms that solve all the remaining problems in polynomial-time. This means that we even know exactly which finite-domain CSPs are in P. Even for the case that $\mathrm{P} = \mathrm{NP}$ holds, the class of finite-domain CSPs is divided by an algebraic borderline whose meaning needs to be further explored.

Motivated not least by questions of practical application, also a theory to study computational complexities of infinite-domain CSPs has been developed. The class of infinite-domain CSPs as a whole is too large to show results of the kind we have in mind; on the

[1] We recommend [AB09] for an introduction to complexity theory.

one hand, as we saw before, every computational problem is polynomial-time equivalent to a (infinite domain) CSP, on the other hand, the universal algebraic approach is not generalizable to all infinite-domain CSPs. Nevertheless, a large class of infinite-domain CSPs has been identified for which both the complexities of the problems are in NP and some of the algebraic methods work. As in the case of finite-domain CSPs, a P vs. NP-complete dichotomy has been conjectured for this class ([BPP19], see also [BOP18], [BKO^{+}19]). The so-called *Bodirsky-Pinsker dichotomy conjecture* has been verified for several restricted subclasses, but is still (wide) open in full generality. There is also again a conjecture about an algebraic borderline that divides the considered class of infinite-domain CSPs into two parts: those that are in P and those that are NP-hard. In this thesis we will prove this conjecture for a certain subclass of CSPs.

2.5 Model Theory

In the study of infinite structures that serve as templates of CSPs, model theory entered the scene along with universal algebra. Model theory studies structures, maps, and sets with the help of logical formulas [Hod97]. The definition of CSPs in terms of primitive positive formulas already indicates how natural it is to integrate model theory into the study of CSPs. We recall some basic definitions from model theory that we need in this thesis.

Let τ be a finite relational signature. The class of finite τ-structures that have an embedding into a τ-structure $\mathfrak{B}$ is called the *age* of $\mathfrak{B}$, denoted by $\mathrm{Age}(\mathfrak{B})$. If $\mathcal{F}$ is a class of finite τ-structures, then $\mathrm{Forb}(\mathcal{F})$ is the class of all finite τ-structures $\mathfrak{A}$ such that no structure from $\mathcal{F}$ embeds into $\mathfrak{A}$. A class $\mathcal{C}$ of finite τ-structures is called *finitely bounded* if there exists a finite set of finite τ-structures $\mathcal{F}$ such that $\mathcal{C} = \mathrm{Forb}(\mathcal{F})$. The structures from $\mathcal{F}$ are called *bounds* or *forbidden substructures*. It is easy to see that a class $\mathcal{C}$ of τ-structures is finitely bounded if and only if it is axiomatisable by a universal first-order τ-sentence. A structure $\mathfrak{B}$ is called finitely bounded if $\mathrm{Age}(\mathfrak{B})$ is finitely bounded.

We will use the following fact about normal representations.

2.21 Proposition (Proposition 2 in [Bod18]) *Let $\mathfrak{B}$ be a normal representation of a finite $\mathbf{A} \in$ RRA. Then the following holds:*

- *$\mathfrak{B}$ is finitely bounded by bounds of size at most three.*
- *The $A_0 \backslash \{\mathrm{Id}\}$-reduct of $\mathfrak{B}$ is finitely bounded by bounds of size at most three.*

Although the definition of homogeneity has already been used in the definition of normal representations, we state it again here. A relational τ-structure $\mathfrak{B}$ is called *homogeneous*, if for every isomorphism between finite substructures of $\mathfrak{B}$ there exists an automorphism of $\mathfrak{B}$ that extends this isomorphism.

2.22 Definition A class of finite τ-structures has the *amalgamation property* if for all structures $\mathfrak{A}, \mathfrak{B}_1, \mathfrak{B}_2 \in \mathcal{C}$ with embeddings $e_1 \colon \mathfrak{A} \to \mathfrak{B}_1$ and $e_2 \colon \mathfrak{A} \to \mathfrak{B}_2$ there exist a structure $\mathfrak{C} \in \mathcal{C}$ and embeddings $f_1 \colon \mathfrak{B}_1 \to \mathfrak{C}$ and $f_2 \colon \mathfrak{B}_2 \to \mathfrak{C}$ such that $f_1 \circ e_1 = f_2 \circ e_2$. If additionally $f_1(B_1) \cap f_2(B_2) = f_1(e_1(A)) = f_2(e_2(A))$, then we say that $\mathcal{C}$ has the *strong amalgamation property*.

Let $\mathfrak{B}_1, \mathfrak{B}_2$ be τ-structures. Then $\mathfrak{B}_1 \cup \mathfrak{B}_2$ is the τ-structure on the domain $B_1 \cup B_2$ such that $R^{\mathfrak{B}_1 \cup \mathfrak{B}_2} := R_1^{\mathfrak{B}} \cup R_2^{\mathfrak{B}}$ for every $R \in \tau$. If Definition 2.22 holds with $\mathfrak{C} := \mathfrak{B}_1 \cup \mathfrak{B}_2$ then we say that $\mathcal{C}$ has the *free amalgamation property*; note that the free amalgamation property implies the strong amalgamation property.

The following theorem, due to Roland Fraïssé connects the amalgamation property with homogeneous structures.

2.23 Theorem ([Fra54, Fra86]; see, e.g., Theorem 6.1.2 in [Hod97]) *Let τ be a finite relational signature and let $\mathcal{C}$ be a class of finite τ-structures that is closed under taking induced substructures and isomorphisms and has the amalgamation property. Then there exists an up to isomorphism unique countable homogeneous structure $\mathfrak{B}$ such that $\mathcal{C} = \mathrm{Age}(\mathfrak{B})$.*

We will use this theorem in Section 4.2.2 to provide normal representations for certain relation algebras.

2.6 Universal Algebra

In this section we present the universal-algebraic approach to the study of CSPs. We start with the basic definitions of clones and polymorphisms. Subsection 2.6.2 serves as an introduction to the universal-algebraic approach to finite-domain CSPs. In the following subsection we look at conservative finite domains and what results were discovered for this special case of structures. Finally, we introduce the universal algebraic methods for infinite structures (Subsection 2.6.4) and consider useful algebraic terminology in the context of normal representations (Subsection 2.6.5).

2.6.1 Clones

In this subsection, we introduce clones and polymorphisms.

2.24 Definition Let B be some set. We denote by $O_B^{(n)}$ the set of all n-ary operations on B and by $O_B := \bigcup_{n \in \mathbb{N}} O_B^{(n)}$ the set of all operations on B. We denote by π_i^j for $i \leqslant j$ the j-ary operation that projects to the i-coordinate, i.e., $\pi_i^j(x_1, \ldots, x_j) = x_i$. A set $\mathscr{C} \subseteq O_B$ is called an *operation clone* on B if it contains all projections of all arities and if it is closed under composition, i.e., for all $f \in \mathscr{C}^{(n)} := \mathscr{C} \cap O_B^{(n)}$ and $g_1, \ldots, g_n \in \mathscr{C}^{(s)} := \mathscr{C} \cap O_B^{(s)}$ it holds that $f(g_1, \ldots, g_n) \in \mathscr{C}$, where $f(g_1, \ldots, g_n)$ is the s-ary function defined as follows

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

We will consider in this thesis operation clones over finite base sets B as well as operation clones over infinite sets B. An operation clone $\mathscr{C}$ on a set B can also be seen as an algebra $\mathbf{B}$ with domain B whose signature consists of the operations of $\mathscr{C}$ such that $f^{\mathbf{B}} := f$ for all $f \in \mathscr{C}$. We write Proj for the operation clone on a two-element set that consists of only the projections.

Operation clones occur naturally as polymorphism clones of relational structures. We explain what this means. If $x_1, \ldots, x_n \in B^k$ and $f \colon B^n \to B$, then we write $f(x_1, \ldots, x_n)$ for the k-tuple obtained by applying f component-wise to the tuples $x_1, \ldots, x_n$.

2.25 Definition Let B be some set. An n-ary operation f on B *preserves* a k-ary relation $R \subseteq B^k$ on B if for all $x_1, \ldots, x_n \in R$ it holds that $f(x_1, \ldots, x_n) \in R$. We denote this by $f \rhd R$.

With the help of the preservation relation $\rhd$ we can define the following two sets.

2.26 Definition Let B be some set, let $\mathcal{R}$ be a set of relation on B and let $\mathscr{F} \subseteq O_B$ be a set of operations on B. Then we define

- $\mathrm{Pol}(\mathcal{R}) := \{f \in O_B \mid \forall R \in \mathcal{R} : f \rhd R\}$,

- $\mathrm{Inv}(\mathscr{F}) := \{R \mid \exists k \in \mathbb{N} : R \subseteq B^k \text{ and } \forall f \in \mathscr{F} : f \rhd R\}$.

We call $\mathrm{Pol}(\mathcal{R})$ the *polymorphisms of* $\mathcal{R}$ and $\mathrm{Inv}(\mathscr{F})$ the *invariant relations under* $\mathscr{F}$.

For a relational τ-structure $\mathfrak{B}$ we define the *polymophism clone of* $\mathfrak{B}$ as

$$\mathrm{Pol}(\mathfrak{B}) := \mathrm{Pol}(\{R^{\mathfrak{B}} \mid R \in \tau\}).$$

This means that a *polymorphism* of $\mathfrak{B}$ is an operation on B that preserves all relations from $\mathfrak{B}$. Since the set of polymorphisms is closed under composition and since a projection is always a polymorphism of $\mathfrak{B}$, it follows that $\mathrm{Pol}(\mathfrak{B})$ is an operation clone on B. In order to provide an additional view on polymorphisms recall the definition of the n-th product structure $\mathfrak{B}^n$. It is easy to see that the n-ary polymorphisms of $\mathfrak{B}$ are precisely the homomorphisms from $\mathfrak{B}^n$ to $\mathfrak{B}$.

2.27 Definition Let $\mathscr{C}$ and $\mathscr{D}$ be operation clones. A function $\mu \colon \mathscr{C} \to \mathscr{D}$ is called a *clone homomorphism* if the following holds:

- μ maps every operation in $\mathscr{C}$ to an operation of the same arity in $\mathscr{D}$.
- μ maps every projection π_i^j in $\mathscr{C}$ to the corresponding projection π_i^j in $\mathscr{D}$.
- preserves the composition of operations, i.e., for every $f \in \mathscr{C}^{(n)}$ and all $g_1, \ldots, g_n \in \mathscr{C}^{(s)}$ the following holds:

$$\mu(f(g_1, \ldots, g_n)) = \mu(f)(\mu(g_1), \ldots, \mu(g_n)).$$

The operation clone O_A on a countable set B can be equipped with the following complete ultrametric d. We first choose an enumeration of B or for better readability assume that $B \subseteq \mathbb{N}$. For two polymorphisms f and g of different arities we define $d(f,g) := 1$. If f and g are both of arity k we have

$$d(f,g) := 2^{-\min\{n \in \mathbb{N} \mid \exists s \in \{1,\ldots,n\}^k : f(s) \neq g(s)\}}.$$

Of course the precise values of d depend on the enumeration that we chose, however all such ultrametrics induce the same topology.

2.28 Definition Let B be a countable set. The topology T_d that is induces by the ultrametric d on O_A is called *topology of pointwise convergence*. A set of operations $\mathscr{C} \subseteq O_A$ is called *closed* if it is closed with respect to T_d.

The following is a straightforward consequence of the definition.

2.29 Lemma (see Proposition 9.5.4 in [Bod21]) *Let $\mathscr{D}$ be an operation clone on B and $\mathscr{C}$ an operation clone on C and let $\nu \colon \mathscr{D} \to \mathscr{C}$ a map. Then ν is uniformly continuous (u.c.) with respect to the ultrametric d if and only if*

$$\forall n \geqslant 1\ \exists \text{ finite } F \subset D \forall f,g \in \mathscr{D}^{(n)} : f|_F = g|_F \Rightarrow \nu(f) = \nu(g).$$

Note that every map between operation clones on finite base sets is uniformly continuous.

2.6.2 The Inv-Pol Galois Connection for Finite Domains

In this section we want to give some insights of how universal algebra was used in the last decades to study computational problems which can be formulated as CSPs. For the sake of a better presentation, we will limit ourselves for now to finite-domain CSPs. Recall that the computational task of $\mathrm{CSP}^{pp}(\mathfrak{B})$ is to decide whether a given primitive positive sentence holds in the fixed relational structure $\mathfrak{B}$. Starting with this definition of CSPs it stands to reason that understanding the complexity of a CSP means to understand which primitive positive sentences hold in a fixed structure $\mathfrak{B}$. That in turn means to understand which primitive positive formulas are satisfiable in $\mathfrak{B}$ or which relations are primitively positively definable in $\mathfrak{B}$. To put it briefly we want to understand $\langle\mathfrak{B}\rangle_{pp}$.

Of course "understanding $\langle\mathfrak{B}\rangle_{pp}$" is a very hazy formulation. The following observation gives an initial impression of what this can mean exactly.

2.30 Proposition ([JCG97], [Jea98]) *Let $\mathfrak{B}_1$ and $\mathfrak{B}_2$ be two structures with finite signatures on the same domain. If $\langle\mathfrak{B}_1\rangle_{pp} \subseteq \langle\mathfrak{B}_2\rangle_{pp}$ holds, then $\mathrm{CSP}^{pp}(\mathfrak{B}_1)$ is logspace reducible to $\mathrm{CSP}^{pp}(\mathfrak{B}_2)$.*

We give a small example of how this proposition can be used to obtain results about the computational complexity of CSPs.

2.31 Example Let $\mathfrak{K}_5$ be the complete graph on 5 vertices and let $\mathfrak{C}_5$ be the undirected cycle of length 5, both on the same domain $\{1,\dots,5\}$. We show that $\langle\mathfrak{K}_5\rangle_{pp} \subseteq \langle\mathfrak{C}_5\rangle_{pp}$. We consider $\mathfrak{K}_5$ as an $\{E\}$-structure and $\mathfrak{C}_5$ as an $\{F\}$-structure. Observe that once we find a primitive positive definition $\varphi_E(x,y)$ of $E^{\mathfrak{K}_5}$ in $\mathfrak{C}_5$ the statement follows, since we can use $\varphi_E(x,y)$ instead of E in every formula that defines a relation from $\langle\mathfrak{K}_5\rangle_{pp}$ and thereby get a definition of the same relation in $\mathfrak{C}_5$. We consider the following formula:

$$\varphi_E(x,y) := \exists z_1, z_2\colon F(x,z_1) \wedge F(z_1,z_2) \wedge F(z_2,y).$$

The formula $\varphi_E(x,y)$ holds for all distinct x and y in $\mathfrak{C}_5$ and does not evaluate to true whenever $x = y$. This implies that $\varphi_E(x,y)$ is a primitive positive definition of $E^{\mathfrak{K}_5}$ and therefore we get that $\langle\mathfrak{K}_5\rangle_{pp} \subseteq \langle\mathfrak{C}_5\rangle_{pp}$ follows. By the previous proposition we get a logspace reduction from $\mathrm{CSP}^{pp}(\mathfrak{K}_5)$ to $\mathrm{CSP}^{pp}(\mathfrak{C}_5)$.

The problem $\mathrm{CSP}(\mathfrak{K}_n)$ is by Example 2.17 well known as the n-coloring problem. It is a classical result that $\mathrm{CSP}(\mathfrak{K}_n)$ is NP-complete for all $n \geqslant 3$ [Kar72] (see also [GJ78]). Combining this with what we observed before, we proved NP-hardness of $\mathrm{CSP}^{pp}(\mathfrak{C}_5)$.

The following classical result gives a description of $\langle\mathfrak{B}\rangle_{pp}$ in terms of polymorphisms and can be seen as the centrepiece of the universal algebraic approach.

2.32 Theorem ([Gei68], [BKKR69]) *Let $\mathfrak{B}$ be a finite τ-structure. Then it holds that*

$$\langle\mathfrak{B}\rangle_{pp} = \mathrm{Inv}(\mathrm{Pol}(\mathfrak{B})).$$

The connection of the two operators Pol and Inv is known as a *Galois connection*. We recommend the book by Pöschel and Kalužnin [PK79] for detailed background information.

Let us continue by taking a closer look at what we obtain from the previous theorem. Assume therefore that we have two finite structures $\mathfrak{B}_1$ and $\mathfrak{B}_2$ with the same domain such that $\mathrm{Pol}(\mathfrak{B}_1) \subseteq \mathrm{Pol}(\mathfrak{B}_2)$, i.e., every polymorphism of $\mathfrak{B}_1$ is also a polymorphism of $\mathfrak{B}_2$. As a direct consequence of the definition of the operator Inv we get that in this case $\mathrm{Inv}(\mathrm{Pol}(\mathfrak{B}_1)) \supseteq \mathrm{Inv}(\mathrm{Pol}(\mathfrak{B}_2))$ holds. The identity from the previous theorem implies that $\langle \mathfrak{B}_1 \rangle_{pp} \supseteq \langle \mathfrak{B}_2 \rangle_{pp}$. As a last step we observe that there exists by Proposition 2.30 a logspace reduction from $\mathrm{CSP}^{pp}(\mathfrak{B}_2)$ to $\mathrm{CSP}^{pp}(\mathfrak{B}_1)$. This implies, for example, that if $\mathrm{CSP}^{pp}(\mathfrak{B}_2)$ is known to be NP-hard, then $\mathrm{CSP}^{pp}(\mathfrak{B}_1)$ is also NP-hard; in other words, this opens up a way to elegantly prove NP-hardness results.

To sum up, we can say that a certain relationship (in this case set-inclusion) of $\mathrm{Pol}(\mathfrak{B}_1)$ and $\mathrm{Pol}(\mathfrak{B}_2)$ controls dependencies of the computational properties between $\mathrm{CSP}^{pp}(\mathfrak{B}_1)$ and $\mathrm{CSP}^{pp}(\mathfrak{B}_2)$. One might wonder what other relationships between operation clones $\mathrm{Pol}(\mathfrak{B}_1)$ and $\mathrm{Pol}(\mathfrak{B}_2)$ lead to complexity results about CSPs. This question was intensively studied in the last 30 years. Many other generalizations of $\mathrm{Pol}(\mathfrak{B}_1) \subseteq \mathrm{Pol}(\mathfrak{B}_2)$ where discovered for example the existence of a clone homomorphism or the satisfiability of certain identities.

We collect some of these results from the last decades that we will use in this thesis. The following theorem is a generalization of Proposition 2.30: It provides a reduction between CSPs, given that the polymorphism clones are in a certain relationship.

2.33 Theorem ([Bir35], [BKJ05]) *Let $\mathfrak{A}$ and $\mathfrak{B}$ be finite structures with finite relational signatures. Then the following are equivalent:*

1. $\mathrm{Pol}(\mathfrak{A})$ *is in* $\mathrm{HSP}^{\mathrm{fin}}(\{\mathrm{Pol}(\mathfrak{B})\})$.
2. *There exists a clone homomorphism* $\mathrm{Pol}(\mathfrak{B}) \to \mathrm{Pol}(\mathfrak{A})$.

If one of these conditions holds, then there exists a logspace reduction from $\mathrm{CSP}(\mathfrak{A})$ *to* $\mathrm{CSP}(\mathfrak{B})$.

With the help of Birkoff's theorem, a famous result in universal algebra, it is easy to show the equivalence of the two conditions. The implication of logspace reduction between the CSPs is proved in [BKJ05]. The proof involves polynomial-time reductions via *primitive positive interpretations of structures*. We will not handle interpretations in this thesis directly, but rather use this theorem (and its corollaries) as a black box result that provides polynomial-time reductions between CSPs.

Recall Example 2.31 from above. In this example we obtained the NP-hardness of $\mathrm{CSP}(\mathfrak{A})$ for a finite relational structure $\mathfrak{A}$ by a combination of a polynomial-time reduction (Proposition 2.30) and a given NP-hard problem (the n-coloring problem). Let us consider another prominent example of an NP-hard CSP.

2.34 Example The relational structure 1IN3 has the domain $\{0,1\}$ and the ternary relation $R^{1\text{IN3}} = \{(1,0,0),(0,1,0),(0,0,1)\}$. The problem $\mathrm{CSP}(1\mathrm{IN3})$ is called the *Positive 1-in-3-3SAT Problem*, which is a well-known NP-complete problem [Sch78].

It is easy to check that $\mathrm{Pol}(1\mathrm{IN3}) = \mathrm{Proj}$ holds. This and the NP-hardness of $\mathrm{CSP}(1\mathrm{IN3})$ in mind, Theorem 2.33 has the following immediate consequence.

2.35 Corollary ([BKJ05]) *Let $\mathfrak{B}$ be a finite structure with finite relational signature. Then the following are equivalent:*

1. $\mathrm{HSP}^{\mathrm{fin}}(\{\mathrm{Pol}(\mathfrak{B})\})$ *contains a 2-element algebra where all operations are projections.*
2. *There exists a clone homomorphism* $\mathrm{Pol}(\mathfrak{B}) \to \mathrm{Proj}$.

If one of these conditions holds, then $\mathrm{CSP}(\mathfrak{B})$ *is NP-complete.*

This corollary provides a criterion for NP-hardness of finite-domain CSPs. We want to remark that the criterion is not a necessary condition for NP-hardness of finite-domain CSPs; there exist examples of finite structure $\mathfrak{B}$ with NP-hard CSPs that do not satisfy the conditions in the theorem. However, a necessary condition (unless $\mathrm{P} = \mathrm{NP}$) can be formulated in terms of *minor-preserving maps* instead of clone homomorphisms. We will not need this here and recommend the interested reader to *the wonderland of reflections* [BOP18]. The reason that we can restrict ourselves to clone homomorphisms (and equivalent conditions) is that in this thesis all polymorphism clones of finite structures are *idempotent*.

2.36 Definition An operation clone $\mathscr{C}$ on a set B. An operation $f \in \mathscr{C}^{(n)}$ is called *idempotent* if for all $x \in B$ it holds that $f(x,\dots,x) = x$. The operation clone $\mathscr{C}$ is called *idempotent* if all its operations are idempotent. We call a structure $\mathfrak{B}$ *idempotent* if $\mathrm{Pol}(\mathfrak{B})$ is idempotent.

The restriction to idempotent finite structures simplifies things a little. In fact, one could consider Corollary 2.35 as an intermediate step on the way to the dichotomy result for idempotent finite structures: the missing part is to find a polynomial-time algorithm for all $\mathrm{CSP}(\mathfrak{B})$ where $\mathrm{Pol}(\mathfrak{B})$ is idempotent and does not satisfy one of the conditions in the theorem. It was a great achievement to turn the absence of clone homomorphisms into a statement about the existence of certain operations in $\mathrm{Pol}(\mathfrak{B})$. This motivates the following definitions.

2.37 Definition Let f be an n-ary operation on set B. Then f is called a *cyclic operation* if

$$\forall x_1,\dots,x_n \in B\colon f(x_1,\dots,x_n) = f(x_n,x_1,\dots,x_{n-1}).$$

The operation f is called a *weak near-unanimity operation* if

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

The operation f is called a *Siggers operation* if

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

With these terms we can formulate the following result, which incorporates work of [Tay77], [MM08], [Sig10] and [BK12].

2.38 Proposition *Let $\mathfrak{B}$ be a structure with finite domain such that* $\mathrm{Pol}(\mathfrak{B})$ *is idempotent. Then the following are equivalent:*

1. $\mathrm{Pol}(\mathfrak{B})$ *contains a Siggers operation.*
2. $\mathrm{Pol}(\mathfrak{B})$ *contains for every prime $p > |B|$ a p-ary cyclic operation.*
3. $\mathrm{Pol}(\mathfrak{B})$ *contains a weak near-unanimity operation.*
4. *There exists no clone homomorphism* $\mathrm{Pol}(\mathfrak{B}) \to \mathrm{Proj}$.

It should be emphasized again that this theorem is so important because it converts the non-existence of a clone homomorphism into the existence of certain operations that can then be used to create polynomial-time algorithms. This is exactly what Bulatov and Zhuk have done independently (in a very sophisticated way, of course) in their proofs of the Feder-Vardi conjecture. Thanks to them, we have the following theorem as a preliminary highlight in the study of CSPs.

2.39 Theorem (Finite-Domain CSP Dichotomy [Bul17], [Zhu17, Zhu20]) *Let $\mathfrak{B}$ be a finite structure with a finite relational signature. If* $\mathrm{Pol}(\mathfrak{B})$ *contains a Siggers operation, then* $\mathrm{CSP}(\mathfrak{B})$ *is in P. Otherwise,* $\mathrm{CSP}(\mathfrak{B})$ *is NP-complete.*

2.6.3 Conservative Domains

In this subsection we consider conservative structures and clones and give some well-known results about them. We call a relational structure $\mathfrak{B}$ *conservative* if it contains all subsets of B as unary relations. Furthermore, an operation $f\colon B^n \to B$ is called *conservative* if for all $x_1, \dots, x_n \in B$ it holds that $f(x_1, \dots, x_n) \in \{x_1, \dots, x_n\}$. A clone is called *conservative* if all its operations are conservative. Note that if $\mathfrak{B}$ is conservative, then $\mathrm{Pol}(\mathfrak{B})$ is conservative, which justifies the similarities in the terminology.

As mentioned before, an operation clone $\mathscr{C}$ on B can be considered as an algebra $\mathbf{B}$ with domain B and an infinite signature. In this sense a conservative operation clone $\mathscr{C}$ on B induces on every set $A \subseteq B$ a subalgebra. It is easy to see that this subalgebra corresponds to an operation clone on A. We call this the *restriction* of $\mathscr{C}$ to A. We later need the following classical result for clones over a two-element set.

2.40 Theorem ([Pos41]) *Let $\mathscr{C}$ be a conservative operation clone on $\{0,1\}$. Then either $\mathscr{C}$ contains only projections, or at least one of the following operations:*

1. *the binary function* min,
2. *the binary function* max,

3. *the minority function,*
4. *the majority function.*

An operation $f\colon B^3 \to B$ is called a *majority operation* if for all $x, y \in B$

$$f(x,x,y) = f(x,y,x) = f(y,x,x) = x$$

holds. It is called a *minority operation* if for all $x, y \in B$

$$f(x,x,y) = f(x,y,x) = f(y,x,x) = y$$

holds.

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

2.41 Definition ([Bul03, Bul11]) 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 $f(a,a) = a$. We say that 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).

According to the previous definition, an "edge type" of a concrete set $\{a,b\} \subseteq B$ is witnessed by a certain operation. For another set $\{c,d\} \subseteq B$ this could a priori be a different operation (even if the two sets have the same edge type). However, Bulatov obtained "uniform witness operations" by the following proposition.

2.42 Proposition (Proposition 3.1.in [Bul11]) *Let $\mathfrak{B}$ be a finite structure. Then there are a binary operation $v \in \mathrm{Pol}(\mathfrak{B})$ and ternary operations $g, h \in \mathrm{Pol}(\mathfrak{B})$ such that for every two-element subset C of B we have that*

- *$v|_C$ is a semilattice operation whenever C has a semilattice edge, and $v|_C(x,y) = x$ otherwise;*
- *$g|_C$ is a majority operation if C is a majority edge, $g|_C(x,y,z) = x$ if C is affine and $g|_C(x,y,z) = v|_C(v|_C(x,y),z)$ 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 $h|_C(x,y,z) = v|_C(v|_C(x,y),z)$ if C has a semilattice edge.*

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.

2.43 Theorem ([Bul03]; see also [Bar11, Bul11, Bul16]) *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 $\{a, b\}$.

2.6.4 Infinite Domains

We saw in the previous paragraph mainly results concerning polymorphism clones on finite sets and CSPs of finite-domain structures, respectively. In fact, there is also a history of research on infinite-domain CSPs. Motivated not least by questions of practical application, a diverse theory of such CSPs developed. For a detailed introduction to the field, we refer to [Bod21]. The class of infinite-domain CSPs as a whole is too large to show results comparable to those from the previous section. A characterization of computational complexity by polymorphisms and their properties reaches its limits. It turns out that a certain kind of finiteness is helpful (and necessary) to get the algebraic machinery working. Accordingly, we focus on structures that have only finitely many definable relations for each arity. Such structures are called *ω-categorical*. We do not want to discuss these structures in detail, but rather make use of the following folklore result.

2.44 Proposition (see, e.g., Lemma 4.3.1 in [Bod21]) *Let τ be a finite relational signature. If $\mathfrak{B}$ is a homogeneous τ-structure, then $\mathfrak{B}$ is ω-categorical.*

Most of the following results were originally shown not for homogeneous structures but for the more general class of ω-categorical structures. However, since we are only dealing with homogeneous infinite structures in this thesis, we take advantage of the previous proposition and quote the results directly in the weaker form. This leads to a reduction of the terminology.

2.45 Theorem ([BN06]) *Let $\mathfrak{B}$ be a homogeneous structure with finite relational signature. Then*

$$\langle \mathfrak{B} \rangle_{pp} = \mathrm{Inv}(\mathrm{Pol}(\mathfrak{B})),$$

i.e., a relation is primitively positively definable in $\mathfrak{B}$ if and only if it is preserved by $\mathrm{Pol}(\mathfrak{B})$.

The following theorem gives a condition for the existence of a reduction from a finite-domain CSP to the CSP of a homogeneous structure. Note that this theorem is inspired by Theorem 2.33. The equivalence of the two statements is proved by a topological variant of Birkoff's theorem in [BP15b] and the statement on the logspace reduction first appeared in the survey article [Bod08] (see Proposition 3 and Theorem 9 therein).

2.46 Theorem ([BP15b], [Bod08]) *Let $\mathfrak{A}$ be a finite and $\mathfrak{B}$ be a homogeneous structure, both with finite relational signatures. Then the following are equivalent:*

1. $\mathrm{Pol}(\mathfrak{A})$ *is in* $\mathrm{HSP}^{\mathrm{fin}}(\{\mathrm{Pol}(\mathfrak{B})\})$.
2. *There exists a uniformly continuous clone homomorphism* $\mathrm{Pol}(\mathfrak{B}) \to \mathrm{Pol}(\mathfrak{A})$.

If one of these conditions holds, then there exists a logspace reduction from $\mathrm{CSP}(\mathfrak{A})$ *to* $\mathrm{CSP}(\mathfrak{B})$.

Analogously to Corollary 2.35, we obtain the following hardness condition for CSPs of homogeneous structures.

2.47 Corollary *Let $\mathfrak{B}$ be a homogeneous structure with finite relational signature. Then the following are equivalent:*

- $\mathrm{HSP}^{\mathrm{fin}}(\{\mathrm{Pol}(\mathfrak{B})\})$ *contains a 2-element algebra where all operations are projections.*
- *There exists a uniformly continuous clone homomorphism* $\mathrm{Pol}(\mathfrak{B}) \to \mathrm{Proj}$.

If one of these conditions holds, then $\mathrm{CSP}(\mathfrak{B})$ *is NP-hard.*

2.6.5 Polymorphisms of Normal Representations

In the following let $\mathbf{A} \in \mathrm{RRA}$ be finite and with normal representation $\mathfrak{B}$. We introduce the following notation, which we will use a lot in this thesis.

2.48 Definition Let $a_1, \ldots, a_n \in A_0$ be atoms of $\mathbf{A}$. Then the $2n$-ary relation $(a_1, \ldots, a_n)^{\mathfrak{B}}$ is defined as follows:

$$(a_1, \ldots, a_n)^{\mathfrak{B}} := \big\{(x_1, \ldots, x_n, y_1, \ldots, y_n) \in B^{2n} \mid \bigwedge_{i \in \{1,\ldots,n\}} a_i^{\mathfrak{B}}(x_i, y_i)\big\}.$$

An operation $f\colon B^n \to B$ is called *edge-conservative* if it satisfies for all $x, y \in B^n$ and all $a_1, \ldots, a_n \in A_0$

$$(a_1, \ldots, a_n)^{\mathfrak{B}}(x, y) \Rightarrow (f(x), f(y)) \in \bigcup_{i \in \{1,\ldots,n\}} a_i^{\mathfrak{B}}.$$

Note that for every $D \subseteq A_0$ the structure $\mathfrak{B}$ contains the relation $\bigcup_{a_i \in D} a_i^{\mathfrak{B}}$. Therefore the next proposition follows immediately from Theorem 2.45, since polymorphisms of $\mathfrak{B}$ preserve all relations of $\mathfrak{B}$.

2.49 Proposition *All polymorphisms of $\mathfrak{B}$ are edge-conservative.*

We end this section with some terminology related to *canonicity*. This will be of great importance in the following, both for polynomial-time algorithms and for NP-hardness results.

2.50 Definition Let $X \subseteq A_0$. An operation $f\colon B^n \to B$ is called *X-canonical* (*with respect to* $\mathfrak{B}$) if there exists a function $\overline{f}\colon X^n \to A_0$ such that for all $a, b \in B^n$ and $O_1, \dots, O_n \in X$, if $(a_i, b_i) \in O_i$ for all $i \in \{1, \dots, n\}$ then $(f(a), f(b)) \in \overline{f}(O_1, \dots, O_n)^{\mathfrak{B}}$. An operation f is called *canonical* (*with respect to* $\mathfrak{B}$) if it is A_0-canonical. In this case we say that the behaviour $\overline{f}$ is *total*. If $X \subsetneq A_0$ we call $\overline{f}$ a *partial behaviour*. The function $\overline{f}$ is called the *behaviour of f on X*. If $X = A_0$ then $\overline{f}$ is just called the *behaviour of f*.

We denote by $\mathrm{Pol}^{\mathrm{can}}(\mathfrak{B})$ the set of all polymorphisms of $\mathfrak{B}$ that are canonical with respect to $\mathfrak{B}$. It will always be clear from the context what the domain of a behaviour $\overline{f}$ is. An operation $f\colon S^2 \to S$ is called *symmetric* if for all $x, y \in S$ it holds that $f(x,y) = f(y,x)$. An X-canonical function f is called *X-symmetric* if the behaviour of f on X is symmetric.

Canonical operations are a central object in the exploration of infinite-domain CSPs and were intensively studied for example in [BPT11], [BM16], [BP21], [BB21], and [MP22]. Usually they are defined in terms of model-theoretic *first-order types*. We would like to note that the canonical operations in our definition are exactly the canonical operations that are usually defined (see, e.g., [Bod21]). This is true because in normal representations the 2-orbits correspond exactly to the set of atoms A_0 of the relation algebra $\mathbf{A}$.

2.51 Proposition *The 2-orbits of the automorphism group* $\mathrm{Aut}(\mathfrak{B})$ *are in 1-to-1 correspondence to the set of atoms A_0 of* $\mathbf{A}$.

Proof: Since $\mathfrak{B}$ is a representation of $\mathbf{A}$, its relations correspond to elements of $\mathbf{A}$. This correspondence preserves the lattice order that is induced by the Boolean algebra part of $\mathbf{A}$. We get that the set-wise minimal, non-empty relations of $\mathfrak{B}$ are precisely the relations $a^{\mathfrak{B}}$ for $a \in A_0$. Since $\mathfrak{B}$ is a normal representation it is by definition square and therefore each 2-tuple is in a relation $a^{\mathfrak{B}}$ for $a \in A_0$. By the homogeneity of $\mathfrak{B}$ it follows that every two 2-tuples that are in the same relation $a^{\mathfrak{B}}$ for $a \in A_0$ can be mapped to each other by an automorphism of $\mathfrak{B}$ and are therefore in the same 2-orbit. $\square$

2.7 Ramsey Theory and Canonisation

We avoid to give an introduction to Ramsey theory, since the only usage of the Ramsey property is via Theorem 2.53, and rather refer to the survey [Bod15] for more details.

Let $\mathfrak{A}$ be a homogeneous τ-structure such that $\mathrm{Age}(\mathfrak{A})$ has the strong amalgamation property. Then the class of all $(\tau \cup \{<\})$-structures $\mathfrak{A}$ such that $<^{\mathfrak{A}}$ is a linear order and whose τ-reduct (i.e., the structure on the same domain, but only with the relations that are denoted by symbols from τ; see, e.g., the book by [Hod97]) is from $\mathrm{Age}(\mathfrak{A})$ is a strong amalgamation class as well (see, e.g., [Bod15]). By Theorem 2.23 there exists an up to isomorphism unique countable homogeneous structure of that age, which we denote by $\mathfrak{A}_<$. It can be shown by a straightforward back-and-forth argument that $\mathfrak{A}_<$ is isomorphic to an expansion of $\mathfrak{A}$, so we identify the domain of $\mathfrak{A}$ and of $\mathfrak{A}_<$ along this isomorphism, and call $\mathfrak{A}_<$ the *expansion of* $\mathfrak{A}$ *by a generic linear order*.

2.52 Theorem ([NR89], see, e.g., Corollary 4.2 in [HN19]) *Let* $\mathfrak{A}$ *be a relational* τ*-structure such that* $\mathrm{Age}(\mathfrak{A})$ *has the free amalgamation property. Then the expansion of* $\mathfrak{A}$ *by a generic linear order has the Ramsey property.*

The following theorem gives a connection of the Ramsey property with the existence of canonical functions and plays a key role in our analysis.

2.53 Theorem ([BPT13], Theorem 5 in [BP21]) *Let* $\mathfrak{B}$ *be a countable homogeneous structure with finite relational signature and the Ramsey property. Let* $h\colon B^k \to B$ *be an operation and let*

$$L := \big\{(x_1, \ldots, x_k) \mapsto \alpha(h(\beta_1(x_1), \ldots, \beta_k(x_k)) \mid \alpha, \beta_1 \ldots, \beta_k \in \mathrm{Aut}(\mathfrak{B})\big\}.$$

Then there exists a canonical operation $g\colon B^k \to B$ *such that for every finite* $F \subset B$ *there exists* $g' \in L$ *such that* $g'|_{F^k} = g|_{F^k}$.

2.54 Remark Let $\mathfrak{A}$ and $\mathfrak{B}$ be homogeneous structures with finite relational signatures. If $\mathfrak{A}$ and $\mathfrak{B}$ have the same domain and the same automorphism group, then $\mathfrak{A}$ has the Ramsey property if and only if $\mathfrak{B}$ has it (see, e.g., [Bod15]).

2.8 Datalog

Let τ be a finite relational signature. In the context of a Datalog program we will call the elements of τ the EDBs (for extensional database predicates). Let ρ be a relational signature disjoint from τ, called IDBs (for intensional database predicates). We assume that ρ contains a symbol *false* of arity 0. A *Datalog rule* is of the form

$$\psi_0 :- \psi_1, \ldots, \psi_n$$

where ψ_0 is an atomic ρ-formula and $\psi_1, \ldots, \psi_n$ are atomic $(\tau \cup \rho)$-formulas. The formula ψ_0 is called the head and the formulas $\psi_1, \ldots, \psi_n$ are called the body of the rule. We assume that every variable from the head also appears in the body. A *Datalog program* Π is a set of Datalog rules. Let $\mathfrak{A}$ be a τ-structure and $\mathfrak{A}'$ a $(\tau \cup \rho)$-expansion of $\mathfrak{A}$. Then $\mathfrak{A}'$ is called a *fixed point of the Datalog program* Π *on* $\mathfrak{A}$ if $\mathfrak{A}'$ satisfies for each rule $\psi_0 :- \psi_1, \ldots, \psi_n$ the sentence

$$\forall \bar{x} \colon (\psi_1 \wedge \ldots \wedge \psi_n) \Rightarrow \psi_0.$$

Let $\mathfrak{A}'$ and $\mathfrak{A}''$ be two $(\tau \cup \rho)$-structures on the same domain. Then $\mathfrak{A}' \cap \mathfrak{A}''$ denotes a $(\tau \cup \rho)$-structure on the same domain, which has for every $R \in (\tau \cup \rho)$ the following definition: $R^{\mathfrak{A}' \cap \mathfrak{A}''} := R^{\mathfrak{A}'} \cap R^{\mathfrak{A}''}$. It can be observed that if both $\mathfrak{A}'$ and $\mathfrak{A}''$ are fixed points of the Datalog program Π on $\mathfrak{A}$, then $\mathfrak{A}' \cap \mathfrak{A}''$ is also a fixed point of the Datalog program Π on $\mathfrak{A}$. Thus, if $\mathfrak{A}$ is finite, there is a unique smallest (with respect to inclusion) fixed point of Π on $\mathfrak{A}$, which we denote by $\Pi(\mathfrak{A})$. Note that the 0-ary predicate *false* behaves in the structure $\Pi(\mathfrak{A})$ like a Boolean value: the 0-ary tuple may or may not be contained in $\mathit{false}^{\Pi(\mathfrak{A})}$. If it is contained we say that Π *derives false on* $\mathfrak{A}$. We denote by $[\![\Pi]\!]$ the class of all finite τ-structures $\mathfrak{A}$ on which Π does not derive *false*. Let $\mathfrak{B}$ be a τ-structure and Π a Datalog program with EDBs τ. We say that the Datalog program Π solves the CSP of $\mathfrak{B}$ if $\mathrm{CSP}(\mathfrak{B}) = [\![\Pi]\!]$. A Datalog program Π has *width* (l, k) if all IDBs are at most l-ary, and if all rules of Π have at most k distinct variables. We also say that in this case Π is an (l, k)-Datalog program.

2.55 Remark Let $\mathbf{A}$ be a relation algebra with a normal representation $\mathfrak{B}$. We will in the following say that an (l, k)-Datalog program Π solves $\mathrm{NSP}(\mathbf{A})$ if the program Π *solves* $\mathrm{CSP}(\mathfrak{B})$. This definition is justified by the correspondence of NSPs and CSPs from Theorem 2.16.

We have already seen in Section 2.6 how universal algebra is used to achieve remarkable results in complexity classification. Also the class of finite-domain CSPs that can be solved by a Datalog program has a beautiful algebraic characterization. In the following presentation this follows from a combination of results from [MM08], [BK09a], and [KKVW15]; for an overview of the proof, see Theorem 8.8.2 in [Bod21].

2.56 Theorem *Let τ be a finite relational signature and let $\mathfrak{B}$ be a finite τ-structure. Then the following statements are equivalent:*

1. *There exist $l \leqslant k$ and an (l, k)-Datalog program Π such that Π 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*

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

There exist many examples of polynomial-time solvable finite-domain CSPs that cannot be solved by a Datalog program. These CSPs encode the problem of solving linear equations over a non-trivial finite abelian group. The criterion from the previous theorem can be used to provide an example of a finite-domain CSP that cannot be solved by any Datalog program.

2.57 Example Consider the structure LIN with domain $\{0, 1\}$, unary relations $\{0\}$ and $\{1\}$ and a ternary relation $R_0^3 = \{(x, y, z) \in \{0, 1\}^3 \mid x + y + z = 0 \pmod 2\}$. One can easily check that LIN does not satisfy Item 2) in Theorem 2.56. It follows that $\mathrm{CSP}(\mathrm{LIN})$ cannot be solved by a Datalog program.

It is known that the logspace reduction from Theorem 2.46 preserves solvability by a Datalog program [ABD09, BPR20, BR22]. This fact, together with the non-solvability by Datalog of $\mathrm{CSP}(1\mathrm{IN}3)$ provides a condition on the CSP of a homogeneous structure that implies the non-existence of a Datalog program that solves the CSP.

2.58 Corollary *Let $\mathfrak{B}$ be a homogeneous structure with finite relational signature. Then the following are equivalent:*

- $\mathrm{HSP}^{\mathrm{fin}}(\{\mathrm{Pol}(\mathfrak{B})\})$ *contains* $\mathrm{Pol}(\mathrm{LIN})$.
- *There exists a uniformly continuous clone homomorphism* $\mathrm{Pol}(\mathfrak{B}) \to \mathrm{Pol}(\mathrm{LIN})$.

If one of these conditions holds, then $\mathrm{CSP}(\mathfrak{B})$ *is not solvable by a Datalog program.*

CHAPTER 3

Hardness Criteria for NSPs

3.1 Introduction

The results of this section are two conditions for a finite relation algebra **A** that imply NP-hardness of the network satisfaction problem of **A**. In order to be able to apply the algebraic theory from Section 2.6 we restrict ourselves to relation algebras with normal representations. Recall that the central object of study in the universal algebraic approach are clones. We focus in this chapter on a particular subset of the clone of polymorphisms of a normal representation $\mathfrak{B}$, namely the group of automorphisms of $\mathfrak{B}$. Since normal representations are by definition homogeneous structures, one can translate back and forth between properties of the relation algebra **A** and properties of $\mathrm{Aut}(\mathfrak{B})$.

For example, $\mathrm{Aut}(\mathfrak{B})$ is primitive if and only if **A** contains no equivalence relation which is different from the *trivial* equivalence relations Id and 1. With this terminology we can rephrase Theorem 1 as follows.

3.1 Theorem *Let* **A** *be a finite relation algebra with a normal representation* $\mathfrak{B}$*. Assume that one of the following holds:*

- $\mathrm{Aut}(\mathfrak{B})$ *preserves an equivalence relation with at least two but finitely many equivalence classes* (*Section 3.2.2*).
- $\mathrm{Aut}(\mathfrak{B})$ *is primitive,* $|B| > 2$ *and* **A** *has a symmetric atom* a *with a forbidden triple* (a, a, a)*, that is,* $a \not\leq a \circ a$ (*Section 3.2.1*)*;*

Then $\mathrm{NSP}(\mathbf{A})$ *is NP-complete.*

We split the proof of this theorem in two different parts. In Section 3.2.1 we prove the statement of the first item and in Section 3.2.2 the second one. Both of our hardness proofs use a technique of factoring $\mathrm{Pol}(\mathfrak{B})$ with respect to an equivalence relation with finitely many classes, and then applying known hardness conditions from corresponding finite-domain constraint satisfaction problems. In Section 3.3 we show how our results can be applied to determine the computational complexity of certain NSPs. This answers some open problems from the literature. Finally, we discuss how these results could form the basis for a general classification programme for network satisfaction problems.

3.2 Hardness Conditions

3.2.1 Finitely Many Equivalence Classes

In the following, $\mathbf{A}$ denotes a finite relation algebra with a normal representation $\mathfrak{B}$.

3.2 Theorem *Suppose that $e \in A$ is such that $e^{\mathfrak{B}}$ is a non-trivial equivalence relation with finitely many classes. Then* $\mathrm{CSP}(\mathfrak{B})$ *is NP-complete.*

Proof: Let $n := 1 \backslash e$ and let $\overline{b}$ denote the equivalence class of any $b \in B$ with respect to $e^{\mathfrak{B}}$. Let $\{c_1, \ldots, c_m\}$ be a set of representatives of the equivalence classes of $e^{\mathfrak{B}}$. A k-ary polymorphism $f \in \mathrm{Pol}(\mathfrak{B})$ induces an operation $\widehat{f}$ of arity k on $C = \{\overline{c_1}, \ldots, \overline{c_m}\}$ in the following way:

$$\widehat{f}(\overline{d_1}, \ldots \overline{d_k}) := \overline{f(d_1, \ldots d_k)}$$

for all $\overline{d_1}, \ldots \overline{d_k} \in \{\overline{c_1}, \ldots, \overline{c_m}\}$. This definition is independent from the choice of the representatives since the polymorphisms preserve the relation $e^{\mathfrak{B}}$. We denote the set of all operations that are induced in this way by operations from $\mathrm{Pol}(\mathfrak{B})$ by $\mathscr{C}$. It is easy to see that $\mathscr{C}$ is an operation clone on a finite set. Moreover, the mapping $\mu \colon \mathrm{Pol}(\mathfrak{B}) \to \mathscr{C}$ defined by $\mu(f) := \widehat{f}$ preserves the arities of operations and maps a projection π_i^j from $\mathrm{Pol}(\mathfrak{B})$ to its corresponding projection in $\mathscr{C}$. We show that μ preserves the composition of operations and choose arbitrary $f \in \mathrm{Pol}^{(n)}(\mathfrak{B})$ and $g_1, \ldots, g_n \in \mathrm{Pol}^{(s)}(\mathfrak{B})$. We get that for all $\overline{d_1}, \ldots \overline{d_k} \in \{\overline{c_1}, \ldots, \overline{c_m}\}$ the following holds:

$$\begin{aligned}
\mu(f(g_1, \ldots, g_n))(\overline{d_1}, \ldots \overline{d_k}) &= \overline{f(g_1, \ldots, g_n)(d_1, \ldots d_k)} \\
&= \overline{f(g_1(d_1, \ldots d_k), \ldots, g_n(d_1, \ldots d_k))} \\
&= \widehat{f}(\overline{g_1(d_1, \ldots d_k)}, \ldots, \overline{g_n(d_1, \ldots d_k)}) \\
&= \widehat{f}(\widehat{g}_1(\overline{d_1}, \ldots \overline{d_k}), \ldots, \widehat{g}_n(\overline{d_1}, \ldots \overline{d_k})) \\
&= \widehat{f}(\widehat{g}_1, \ldots, \widehat{g}_n)(\overline{d_1}, \ldots \overline{d_k}) \\
&= \mu(f)(\mu(g_1), \ldots, \mu(g_n))(\overline{d_1}, \ldots \overline{d_k}).
\end{aligned}$$

This proves that μ is a clone homomorphism. To show that μ is uniformly continuous, we use Lemma 2.29; it suffices to observe that if two k-ary operations $f, g \in \mathrm{Pol}(\mathfrak{B})$ are equal on $F := \{c_1, \ldots, c_m\}$, then they induce the same operation on the equivalence classes.

Suppose for contradiction that $\mathscr{C}$ contains a p-ary cyclic operation for every prime $p > m$. Case 1: $m = 2$. By assumption there exists a ternary cyclic operation $\widehat{f} \in \mathscr{C}$. Since $e^{\mathfrak{B}}$ is non-trivial, one of the equivalence classes of $e^{\mathfrak{B}}$ must have size at least two. So we may without loss of generality assume that $\overline{c_1}$ contains at least two elements. Let $c_1' \in \overline{c_1}$ with $c_1 \neq c_1'$. We have that $\overline{f(c_1, c_1, c_2)} = \overline{f(c_2, c_1, c_1)}$ which means that

$$\big(f(c_1, c_1, c_2), f(c_2, c_1, c_1)\big) \in e^{\mathfrak{B}}. \tag{3.1}$$

On the other hand $(n, \mathrm{Id}, n)^{\mathfrak{B}}\big((c_1, c_1, c_2), (c_2, c_1, c_1)\big)$. Since f is an edge conservative polymorphism we have that

$$\big(f(c_1, c_1, c_2), f(c_2, c_1, c_1)\big) \in (n \cup \mathrm{Id})^{\mathfrak{B}}. \tag{3.2}$$

Combining (3.1) and (3.2) we obtain that

$$f(c_1, c_1, c_2) = f(c_2, c_1, c_1). \tag{3.3}$$

Similarly, $\overline{f(c_2, c_1, c_1)} = \overline{f(c_1, c_2, c_1)}$. Since f preserves the equivalence relation $e^{\mathfrak{B}}$ we also have $\big(f(c_1, c_2, c_1), f(c_1', c_2, c_1)\big) \in e^{\mathfrak{B}}$. But then $(f(c_2, c_1, c_1), f(c_1', c_2, c_1)) \in e^{\mathfrak{B}}$ holds. Also note that $(n, n, \mathrm{Id})^{\mathfrak{B}}\big((c_2, c_1, c_1), (c_1', c_2, c_1)\big)$ implies that $\big(f(c_2, c_1, c_1), f(c_1', c_2, c_1)\big) \in (n \cup \mathrm{Id})^{\mathfrak{B}}$. These two facts together imply $f(c_2, c_1, c_1) = f(c_1', c_2, c_1)$. By (3.3) and the transitivity of equality we get $f(c_1, c_1, c_2) = f(c_1', c_2, c_1)$. But this is impossible because $(e, n, n)^{\mathfrak{B}}\big((c_1, c_1, c_2), (c_1', c_2, c_1)\big)$ implies that $f(c_1, c_1, c_2) \neq f(c_1', c_2, c_1)$.

<u>Case 2:</u> $m > 2$. Let f be a p-ary cyclic operation for some prime $p > m$. Consider the representatives c_1, c_2 and c_3. By the cyclicity of $\widehat{f}$ we have

$$\overline{f(c_1, c_2, \ldots, c_1, c_2, c_3)} = \overline{f(c_3, c_1, c_2 \ldots, c_1, c_2)}$$

and therefore

$$\big(f(c_1, c_2, \ldots, c_1, c_2, c_3), f(c_3, c_1, c_2 \ldots, c_1, c_2)\big) \in e^{\mathfrak{B}}. \tag{3.4}$$

On the other hand,

$$(n, n, n, \ldots, n, n)^{\mathfrak{B}}\big((c_1, c_2, \ldots, c_1, c_2, c_3), (c_3, c_1, c_2 \ldots, c_1, c_2)\big)$$

and since f preserves $n^{\mathfrak{B}}$ we get that

$$\big(f(c_1, c_2, \ldots, c_1, c_2, c_3), f(c_3, c_1, c_2 \ldots, c_1, c_2)\big) \in n^{\mathfrak{B}},$$

contradicting (3.4).

We showed that there exists a prime $p > m$ such that $\mathscr{C}$ does not contain a p-ary cyclic polymorphism and therefore Proposition 2.38 implies the existence of a clone homomorphism $\nu\colon \mathscr{C} \to \mathrm{Proj}$. Note that by Lemma 2.29 it follows that clone homomorphism between clones on finite base sets are always uniformly continuous.

Since the composition of uniformly continuous minor-preserving maps is again uniformly continuous and a clone homomorphism, there exists a uniformly continuous clone homomorphism map $\nu \circ \mu\colon \mathrm{Pol}(\mathfrak{B}) \to \mathrm{Proj}$. This map implies the NP-hardness of $\mathrm{CSP}(\mathfrak{B})$ by Theorem 2.47. $\square$

3.2.2 No Non-Trivial Equivalence Relations

In this section $\mathbf{A}$ denotes a finite relation algebra with a normal representation $\mathfrak{B}$ with $|B| > 2$.

3.3 Definition The automorphism group $\mathrm{Aut}(\mathfrak{C})$ of a relational structure $\mathfrak{C}$ is called *primitive* if $\mathrm{Aut}(\mathfrak{C})$ does not preserve a non-trivial equivalence relation, i.e., the only equivalence relations that are preserved by $\mathrm{Aut}(\mathfrak{C})$ are Id and C^2.

3.4 Proposition *Let a be an atom of $\mathbf{A}$. If $\mathrm{Aut}(\mathfrak{B})$ is primitive then $a \subseteq \mathrm{Id}$ implies $a = \mathrm{Id}$.*

Proof: If $a \subsetneq \mathrm{Id}$ then

$$c := \mathrm{Id} \cup (a \circ 1 \circ a)$$

would be such that $c^{\mathfrak{B}}$ is a non-trivial equivalence relation. □

3.5 Proposition *Let a be a symmetric atom of $\mathbf{A}$ with $a \cap \mathrm{Id} = 0$. If $\mathrm{Aut}(\mathfrak{B})$ is primitive then $a^{\mathfrak{B}} \circ a^{\mathfrak{B}} \neq \mathrm{Id}$.*

Proof: Assume for contradiction $a^{\mathfrak{B}} \circ a^{\mathfrak{B}} = \mathrm{Id}^{\mathfrak{B}}$. This implies $(\mathrm{Id} \cup a)^{\mathfrak{B}} \circ (\mathrm{Id} \cup a)^{\mathfrak{B}} \subset (\mathrm{Id} \cup a)^{\mathfrak{B}}$ and therefore $(\mathrm{Id} \cup a)^{\mathfrak{B}}$ is an equivalence relation. Since $\mathfrak{B}$ is primitive $(\mathrm{Id} \cup a)^{\mathfrak{B}} = B^2$. By assumption B contains at least 3 elements. These elements are now all connected by the atomic relation $a^{\mathfrak{B}}$. This is a contradiction to our assumption $a^{\mathfrak{B}} \circ a^{\mathfrak{B}} = \mathrm{Id}^{\mathfrak{B}}$. □

Higman's lemma states that a permutation group G on a set B is primitive if and only if for every two distinct elements $x, y \in B$ the undirected graph with vertex set B and edge set $\{\{\alpha(x), \alpha(y)\} \mid \alpha \in G\}$ is connected (see, e.g., [Cam99]). We need the following variant of this result for $\mathrm{Aut}(\mathfrak{B})$; we also present its proof since we are unaware of any reference in the literature. If $a \in A$ then a sequence $(b_0, \ldots, b_n) \in B^{n+1}$ is called an *a-walk* (*of length n*) if $(b_i, b_{i+1}) \in a^{\mathfrak{B}}$ for every $i \in \{0, \ldots, n-1\}$ (we count the number of traversed edges rather than the number of vertices when defining the length).

3.6 Lemma *Let $a \in A$ be a symmetric atom of $\mathbf{A}$ with $a \cap \mathrm{Id} = 0$ and suppose that $\mathrm{Aut}(\mathfrak{B})$ is primitive. Then there exists an $a^{\mathfrak{B}}$-walk of even length between any $x, y \in B$. Moreover, there exists $k \in \mathbb{N}$ such that for all $x, y \in B$ there exists an $a^{\mathfrak{B}}$-walk of length $2k$ between x and y.*

Proof: If R is a binary relation then $R^k = R \circ R \circ \cdots \circ R$ denotes the k-th relational power of R. The sequence of binary relations $L_n := \mathrm{Id}^{\mathfrak{B}} \cup \bigcup_{k=1}^{n} (a^{\mathfrak{B}})^{2k}$ is non-decreasing by definition and terminates because all binary relations are unions of at most finitely many atoms. Therefore, there exists $k \in \mathbb{N}$ such for all $n \geqslant k$ we have $L_n = L_k$. Note that L_k is an equivalence relation, namely the relation "there exists an $a^{\mathfrak{B}}$-walk of even length between x and y". Since $\mathfrak{B}$ is primitive L_k must be trivial. If $L_k = B^2$ then there exists an $a^{\mathfrak{B}}$-walk of length $2k$ between any two $x, y \in B$ and we are done. Otherwise,

$$L_k = \{(x, x) \mid x \in B\} = \mathrm{Id}^{\mathfrak{B}}.$$

Since a is symmetric $a^{\mathfrak{B}} \circ a^{\mathfrak{B}} \neq 0$ and $a^{\mathfrak{B}} \circ a^{\mathfrak{B}}$ contains therefore an atom. But then $a^{\mathfrak{B}} \circ a^{\mathfrak{B}} \subseteq L_k$ implies by Proposition 3.4 $a^{\mathfrak{B}} \circ a^{\mathfrak{B}} = L_k$. This is a contradiction to Proposition 3.5. □

3.7 Lemma *Let $a \in A$ be a symmetric atom of* $\mathbf{A}$ *such that* $\mathrm{Aut}(\mathfrak{B})$ *is primitive and* (a,a,a) *is forbidden. Then all polymorphisms of* $\mathfrak{B}$ *are* $\{\mathrm{Id}, a\}$*-canonical.*

In the proof, we need the following notation. Let $a_1, \ldots, a_k \in A$ be such that $a_1 = \ldots = a_j$ and $a_{j+1} = \ldots = a_k$. Instead of writing $(a_1, \ldots, a_n)^{\mathfrak{B}}$ we use the shortcut $(a_1|_j a_{j+1})^{\mathfrak{B}}$.

Proof: (*of Lemma 3.7*) The following ternary relation R on B is primitive positive definable in $\mathfrak{B}$.

$$R := \left\{(x_1, x_2, x_3) \in B^3 \mid (a \cup \mathrm{Id})^{\mathfrak{B}}(x_1, x_2) \wedge (a \cup \mathrm{Id})^{\mathfrak{B}}(x_2, x_3) \wedge a^{\mathfrak{B}}(x_1, x_3)\right\}$$

Observe that $c \in R$ if and only if $a^{\mathfrak{B}}(c_1, c_2) \wedge \mathrm{Id}^{\mathfrak{B}}(c_2, c_3)$ or $\mathrm{Id}^{\mathfrak{B}}(c_1, c_2) \wedge a^{\mathfrak{B}}(c_2, c_3)$.

Let f be a polymorphism of $\mathfrak{B}$ of arity n. Let $x, y, u, v \in B^n$ be arbitrary such that (x, y) and (u, v) have the same $\{\mathrm{Id}, a\}$-configuration. Without loss of generality we may assume that $(a|_j \mathrm{Id})^{\mathfrak{B}}(x, y)$ and $(a|_j \mathrm{Id})^{\mathfrak{B}}(u, v)$. Now consider $p, q \in B^n$ such that $(\mathrm{Id}\,|_j a)^{\mathfrak{B}}(p, q)$ holds.

Note that by the edge-conservativeness of f the following holds:

$$(f(x), f(y)) \in (a \cup \mathrm{Id})^{\mathfrak{B}}, (f(u), f(v)) \in (a \cup \mathrm{Id})^{\mathfrak{B}} \text{ and } (f(p), f(q)) \in (a \cup \mathrm{Id})^{\mathfrak{B}}.$$

By Lemma 3.6 there exists a $k \in \mathbb{N}$ such that for every $i \in \{1, \ldots, n\}$ there exists an $a^{\mathfrak{B}}$-walk $(s_i^0, \ldots, s_i^k)$ with $s_i^0 = y_i$ and $s_i^k = p_i$. Now consider the following walk in B^n:

$$\begin{aligned}
&(a|_j \mathrm{Id})^{\mathfrak{B}}(x, y)\\
&(\mathrm{Id}\,|_j a)^{\mathfrak{B}}\left(y, (s_1^0, \ldots s_j^0, s_{j+1}^1, \ldots s_n^1)\right)\\
&(a|_j \mathrm{Id})^{\mathfrak{B}}\left((s_1^0, \ldots s_j^0, s_{j+1}^1, \ldots s_n^1), (s_1^1, \ldots s_j^1, s_{j+1}^1, \ldots s_n^1)\right)\\
&\quad\vdots\\
&(a|_j \mathrm{Id})^{\mathfrak{B}}((s_1^i, \ldots s_j^i, s_{j+1}^{i+1}, \ldots s_n^{i+1}), (s_1^{i+1}, \ldots s_j^{i+1}, s_{j+1}^{i+1}, \ldots s_n^{i+1}))\\
&(\mathrm{Id}\,|_j a)^{\mathfrak{B}}((s_1^{i+1}, \ldots s_j^{i+1}, s_{j+1}^{i+1}, \ldots s_n^{i+1}), (s_1^{i+1}, \ldots s_j^{i+1}, s_{j+1}^{i+2}, \ldots s_n^{i+2}))\\
&\quad\vdots\\
&(a|_j \mathrm{Id})^{\mathfrak{B}}((s_1^{k-1}, \ldots, s_j^{k-1}, s_{j+1}^k, \ldots, s_n^k), p)\\
&(\mathrm{Id}\,|_j a)^{\mathfrak{B}}(p, q)
\end{aligned}$$

Every three consecutive elements on this walk are component wise in the relation R. Since R is primitive positive definable the polymorphism f preserves R by Theorem 2.45. This means that f maps this walk on a walk where the atomic relations are an alternating sequence of $a^{\mathfrak{B}}$ and $\mathrm{Id}^{\mathfrak{B}}$, which implies

$$(f(x), f(y)) \in a^{\mathfrak{B}} \Leftrightarrow (f(p), f(q)) \in \mathrm{Id}^{\mathfrak{B}}.$$

If we repeat the same argument with a walk from q to v we get:

$$(f(p), f(q)) \in a^{\mathfrak{B}} \Leftrightarrow (f(u), f(v)) \in \mathrm{Id}^{\mathfrak{B}}.$$

Combining these two equivalences gives us

$$(f(x), f(y)) \in a^{\mathfrak{B}} \Leftrightarrow (f(u), f(v)) \in a^{\mathfrak{B}}.$$

Since the tuples $x, y, u, v \in B^n$ were arbitrary this shows that f is $\{\mathrm{Id}, a\}$-canonical. □

3.8 Theorem *Let* $\mathrm{Aut}(\mathfrak{B})$ *be primitive and let a be a symmetric atom of* $\mathbf{A}$ *such that* (a, a, a) *is forbidden. Then* $\mathrm{CSP}(\mathfrak{B})$ *is NP-hard.*

Proof: By Lemma 3.7 we know that all polymorphisms of $\mathfrak{B}$ are $\{a, \mathrm{Id}\}$-canonical. This means that every $f \in \mathrm{Pol}(\mathfrak{B})$ induces an operation $\overline{f}$ of the same arity on the set $\{a, \mathrm{Id}\}$. Let $\mathscr{C}_2$ be the set of induced operations. Note that $\mathscr{C}_2$ is an operation clone on a Boolean domain. The mapping $\mu\colon \mathrm{Pol}(\mathfrak{B}) \to \mathscr{C}_2$ defined by $\mu(f) := \overline{f}$ is a uniformly continuous clone homomorphism.

Assume for contradiction that there exists a ternary cyclic polymorphism $\overline{s}$ in $\mathscr{C}_2$. Let $x, y, z \in B^3$ be such that

$$\begin{aligned} &(a, a, \mathrm{Id})^{\mathfrak{B}}(x, y), \\ &(\mathrm{Id}, a, a)^{\mathfrak{B}}(y, z), \\ \text{and } &(a, \mathrm{Id}, a)^{\mathfrak{B}}(x, z). \end{aligned}$$

By the cyclicity of the operation $\overline{s}$ and the edge-conservativeness of s we have that either

$$(s(x), s(y)) \in a^{\mathfrak{B}}, (s(y), s(z)) \in a^{\mathfrak{B}} \text{ and } (s(x), s(z)) \in a^{\mathfrak{B}}$$

or

$$(s(x), s(y)) \in \mathrm{Id}^{\mathfrak{B}}, (s(y), s(z)) \in \mathrm{Id}^{\mathfrak{B}} \text{ and } (s(x), s(z)) \in \mathrm{Id}^{\mathfrak{B}}.$$

Since (a, a, a) is forbidden, the second case holds. Note that $\mathbf{A}$ must have an atom $b \neq \mathrm{Id}$ such that the triple (a, a, b) is allowed, because otherwise a would be an equivalence relation. Now consider $u, v, w \in B^3$ such that

$$(a, a, \mathrm{Id})^{\mathfrak{B}}(u, v),$$

$$\begin{aligned} &(\mathrm{Id}, a, a)^{\mathfrak{B}}(v, w), \\ \text{and } &(a, b, a)^{\mathfrak{B}}(u, w). \end{aligned}$$

Since s is $\{a, \mathrm{Id}\}$-canonical and with the observation from before we have

$$(s(u), s(v)) \in \mathrm{Id}^{\mathfrak{B}} \text{ and } (s(v), s(w)) \in \mathrm{Id}^{\mathfrak{B}}.$$

Now the transitivity of equality contradicts $(s(u), s(w)) \in (a \cup b)^{\mathfrak{B}}$.

We conclude that $\mathscr{C}_2$ does not contain a ternary cyclic operation. Since the domain of $\mathscr{C}_2$ has size two, Proposition 2.38 implies the existence of a uniformly continuous clone homomorphism $\nu \colon \mathscr{C}_2 \to \mathrm{Proj}$. The composition $\nu \circ \mu \colon \mathrm{Pol}(\mathfrak{B}) \to \mathrm{Proj}$ is also a uniformly continuous clone homomorphism and therefore by Theorem 2.47 the problem $\mathrm{CSP}(\mathfrak{B})$ is NP-hard. □

$\circ$	Id	a	b
Id	Id	a	b
a	a	$\neg b$	b
b	b	b	$\neg b$

$\circ$	Id	a	b
Id	Id	a	b
a	a	$\neg a$	$0'$
b	b	$0'$	1

Figure 3.1: Multiplication tables of relation algebras #13 (left) and #17 (right).

3.3 Examples and Discussion

3.3.1 Examples

Andréka and Maddux classified *small relation algebras*, i.e., finite relation algebras with at most 3 atoms [AM94]. We consider the complexity of the network satisfaction problem of two of them, namely the relation algebras #13 and #17 (we use the enumeration from [AM94]). Both relation algebras have normal representations (see below) and fall into the scope of our hardness criteria. Cristani and Hirsch [CH04] classified the complexities of the network satisfaction problems for small relation algebras, but due to a mistake the algebras #13 and #17 were left open.

3.9 Example (Relation Algebra #13) The relation algebra #13 is given by the multiplication table in Fig. 4.1. This finite relation algebra has a normal representation $\mathfrak{B}$ defined as follows. Let V_1 and V_2 be countable, disjoint sets. We set $B := V_1 \cup V_2$ and define the following atomic relations:

$$\mathrm{Id}^{\mathfrak{B}} := \{(x,x) \in B^2\},$$
$$a^{\mathfrak{B}} := \{(x,y) \in B^2 \setminus \mathrm{Id}^{\mathfrak{B}} \mid (x \in V_1 \wedge y \in V_1) \vee (x \in V_2 \wedge y \in V_2)\},$$
$$b^{\mathfrak{B}} := \{(x,y) \in B^2 \setminus \mathrm{Id}^{\mathfrak{B}} \mid (x \in V_1 \wedge y \in V_2) \vee (x \in V_2 \wedge y \in V_1)\}.$$

It is easy to check that this structure is a square representation for #13. Moreover, this structure is fully universal for #13 and homogeneous, and therefore a normal representation.

Note that the relation $(\mathrm{Id} \cup a)^{\mathfrak{B}}$ is an equivalence relation where V_1 and V_2 are the two equivalence classes. Therefore we get by Theorem 3.2 that the (general) network satisfaction problem for the relation algebra #13 is NP-hard. We mention that this result can also be deduced from the results in [BMPP19].

3.10 Example (Relation algebra #17) The relation algebra #17 is given by the multiplication table in Figure 4.1. Let $\mathfrak{N} = (V; E^{\mathfrak{N}})$ be the countable, homogeneous, universal triangle-free, undirected graph (see [Hod97]), also called called a *Henson graph*. We use this Henson graph to obtain a square representation $\mathfrak{B}$ with domain V for the relation algebra #17 as follows:

$$\mathrm{Id}^{\mathfrak{B}} := \{(x,x) \in V^2\},$$

$$a^{\mathfrak{B}} := \{(x,y) \in V^2 \mid (x,y) \in E^{\mathfrak{N}}\},$$
$$b^{\mathfrak{B}} := \{(x,y) \in B^2 \setminus \mathrm{Id}^{\mathfrak{B}} \mid (x,y) \notin E^{\mathfrak{N}}\}.$$

This structure is homogeneous and fully universal since $\mathfrak{N}$ is homogeneous and embeds every triangle free graph. This implies that $\mathfrak{B}$ is a normal representation of relation algebra #17. It is easy to see that there exists no non-trivial equivalence relation in this relation algebra. For the atom a the triangle (a,a,a) is forbidden, which means we can apply Theorem 3.8 and get NP-hardness for the (general) network satisfaction problem for the relation algebra #17. Also in this case, the hardness result can also be deduced from the results in [BMPP19].

3.3.2 Conclusion and Future Work

Both of our criteria, Theorem 3.2 and Theorem 3.8, show the NP-hardness for relatively large classes of finite relation algebras. In Section 3.3.1 we applied these results to settle the complexity status of two problems that were left open in [CH04].

To obtain our general hardness conditions we used the universal algebraic approach for studying the complexity of constraint satisfaction problems. This approach may lead to a solution of Hirsch's RBCP for all finite relation algebras $\mathbf{A}$ with a normal representation $\mathfrak{B}$. It is also relatively easy to prove that the network satisfaction problem for $\mathbf{A}$ is NP-complete if $\mathfrak{B}$ has an equivalence relation with an equivalence class of finite size larger than two. Hence, the next steps that have to be taken in this direction are the following.

- Classify the complexity of the network satisfaction problem for finite relation algebras $\mathbf{A}$ where the normal representation has a primitive automorphism group.
- Classify the complexity of the network satisfaction problem for relation algebras that have equivalence relations with infinitely many classes of size two.
- Classify the complexity of the network satisfaction problem for relation algebras that have equivalence relations with infinitely many infinite classes.

CHAPTER 4

Symmetric Relation Algebras with a Flexible Atom

4.1 Introduction

More than two decades ago, Hirsch [Hir96] asked the *Really Big Complexity Problem* (*RBCP*): can we classify the computational complexity of the network satisfaction problem for every finite relation algebra? For example, the complexity of the network satisfaction problem for the Point Algebra and the Left Linear Point Algebra is in P [VKv90, BK07], while it is NP-complete for Allen's Interval Algebra, RCC5, and RCC8 [All83, RN99].

We already mentioned that there also exist relation algebras where the complexity of the network satisfaction problem is not in NP: Hirsch constructed an example of a finite relation algebra which has an undecidable network satisfaction problem [Hir99]. This result might be surprising at first sight; it is related to the fact that the representation of a finite relation algebra by concrete binary relations over some set can be quite complicated. We also mention that not every finite relation algebra has a representation [Lyn50]. There are even non-representable relation algebras that are symmetric [Mad06a]; a relation algebra is *symmetric* if every element is its own converse.

A simple condition that implies that a finite relation algebra $\mathbf{A}$ has a representation is the existence of a so-called *flexible atom* [Com84, Mad82]. A flexible atom is an element of $\mathbf{A}$ that is maximally unconstrained in its interaction with other elements of the relation algebra; the formal definitions can be found in Section 4.2.

Relation algebras with a flexible atom have been studied intensively, for example in the context of the *flexible atoms conjecture* [Mad94, AMM08]. We will see that integral relation algebras with a flexible atom even have a normal representation, recall that this is a representation which is fully universal, square, and homogeneous (see Definition 2.15). The network satisfaction problem for a relation algebra with a normal representation can be seen as a constraint satisfaction problem for an infinite structure $\mathfrak{B}$ that is well-behaved from a model-theoretic point of view; in particular, we may choose $\mathfrak{B}$ to be homogeneous and finitely bounded.

Constraint satisfaction problems over finite domains have been studied intensively in the past two decades, and tremendous progress has been made concerning systematic findings about their computational complexity. As a highlighting result, [Bul17] and [Zhu17, Zhu20] proved the famous Feder-Vardi dichotomy conjecture which states that every finite-domain CSP is in P or NP-complete. Both proofs build on an important connection between the computational complexity of constraint satisfaction problems and universal algebra (Section 2.6.2).

The universal-algebraic approach can also be applied to study the computational complexity of countably infinite homogeneous structures $\mathfrak{B}$ with finite relational signature [BN06] (Section 2.6.4). If $\mathfrak{B}$ is finitely bounded, then $\mathrm{CSP}(\mathfrak{B})$ is contained in NP(see, e.g. [Bod12]). If $\mathfrak{B}$ is homogeneous and finitely bounded then a complexity dichotomy has been conjectured, along with algebraic criteria that distinguish NP-complete from polynomial-time solvable problems [BPP19]. The exact formulation of the conjecture from [BPP19] in full generality requires concepts that we do not need to prove our results. In Theorem 4.45 we verify these conjectures for all normal representations of finite integral symmetric relation algebras with a flexible atom, and thereby also solve Hirsch's RBCP for symmetric relation algebras with a flexible atom. Phrased in the terminology of relation algebras, our result is the following.

4.1 Theorem *Let* $\mathbf{A}$ *be a finite symmetric relation algebra with a flexible atom, and let* A_0 *be the set of atoms of* $\mathbf{A}$. *Then one of the following holds:*

- *There exists an operation* $f\colon A_0^6 \to A_0$ *such that*
  1. f *preserves the allowed triples of* $\mathbf{A}$,
  2. $\forall x_1, \dots, x_6 \in A_0.\ f(x_1, \dots, x_6) \in \{x_1, \dots x_6\}$,
  3. f *satisfies the Siggers identity:* $\forall x, y, z \in A_0.\ f(x, x, y, y, z, z) = f(y, z, x, z, x, y)$,

  in this case the network satisfaction problem for $\mathbf{A}$ *is in P.*

- *The network satisfaction problem for* $\mathbf{A}$ *is NP-complete.*

Moreover, the satisfiability of the Siggers identity in Theorem 4.1 is a decidable criterion for $\mathbf{A}$ that is a sufficient condition for the polynomial-time tractability of the network satisfaction problem of $\mathbf{A}$. We want to mention that there are several other equivalent criteria that could be used instead of the first item in the theorem, namely all characterizations of Taylor Algebras for finite conservative algebras (see, e.g., [Bul03]).

The content of this chapter consists largely of the article

[BK21] Manuel Bodirsky and Simon Knäuer, *Network satisfaction for symmetric relation algebras with a flexible atom*, Proceedings of the AAAI Conference on Artificial Intelligence **35** (2021), no. 7, 6218–6226.

4.1.1 Proof Strategy

Every finite integral representable relation algebra $\mathbf{A}$ with a flexible atom has a normal representation $\mathfrak{B}$; for completeness, and since we are not aware of a reference for this fact, we include a proof in Section 4.2. It follows that the classification question about the complexity of the network satisfaction problem of $\mathbf{A}$ can be translated into a question about the complexity of the constraint satisfaction problem for the relational structure $\mathfrak{B}$.

We then associate a finite relational structure $\mathfrak{O}$ to $\mathfrak{B}$ and show that $\mathrm{CSP}(\mathfrak{B})$ can be reduced to $\mathrm{CSP}(\mathfrak{O})$ in polynomial-time (Section 4.3). If the structure $\mathfrak{O}$ satisfies the condition of the first case in Theorem 4.1, then known results about finite-domain CSPs imply that $\mathrm{CSP}(\mathfrak{O})$ is in P [Bul03, Bul16, Bar11], and hence $\mathrm{CSP}(\mathfrak{B})$ is in P, too. If the first case in Theorem 4.1 does not apply, then known results about finite-domain algebras imply that there are $a, b \in A_0$ such that the *canonical polymorphisms* of $\mathfrak{B}$ act as a projection on $\{a, b\}$ [Bul03, Bul11, Bul16, Bar11]. We first observe NP-hardness of $\mathrm{CSP}(\mathfrak{B})$ if $\mathfrak{B}$ does not have a binary injective polymorphism (Section 4.5). If $\mathfrak{B}$ has a binary injective polymorphism, we use results from structural Ramsey theory to show that $\mathfrak{B}$ must even have a binary injective polymorphism which is canonical (Section 4.6). This implies that none of a, b equals $\mathrm{Id} \in A$. We then prove that $\mathfrak{B}$ does not have a binary $\{a, b\}$-symmetric polymorphism; also in this step, we apply Ramsey theory. In Section 4.7 we show that this in turn implies that all polymorphisms of $\mathfrak{B}$ must be canonical on $\{a, b\}$. Finally, we show that $\mathfrak{B}$ cannot have a polymorphism which acts as a majority or as a minority on $\{a, b\}$, and thus by Schaefer's theorem all polymorphisms of $\mathfrak{B}$ act as a projection on $\{a, b\}$. This is again implied by results from Section 4.6. Finally it follows that $\mathrm{CSP}(\mathfrak{B})$ is NP-hard. This concludes the proof of Theorem 4.1.

Our proof follows a strategy that was applied several times in the study of infinite-domain constraint satisfaction problems and recently described and generalized by [MP22]. We give some details about this in Section 4.9.

4.1.2 Organisation of Chapter 4

The basic concepts and tools that are used in this part were already introduced in Chapter 2. In Section 4.2 we define flexible atoms and obtain first results about representable relation algebras with a flexible atom. Section 4.3 is dedicated to the atom structure and the polynomial-time tractability results. In Section 4.4 we provide an additional perspective on the class of computational problems under consideration; in this section we define those problems completely without the use of the relation algebra framework. The reader can get better intuition of the class of problems studied in this article, however our results and proofs do not rely on that section. The Sections 4.5-4.7 contain the main parts of the proof as outlined in the previous paragraph. In Section 4.8 we put everything together and prove the main theorem. We end with a conclusion and a small discussion of our result.

4.2 Relation Algebras with a Flexible Atom

In this section we define the concept of a flexible atom and show how to reduce the classification problem for the network satisfaction problem for a finite $\mathbf{A} \in \mathrm{RRA}$ with a flexible atom to the situation where $\mathbf{A}$ is additionally integral (Proposition 4.4). A finite relation algebra $\mathbf{A}$ is called *integral* if the element Id is an atom of $\mathbf{A}$ (cf. [Mad06b]).

Then we show that an integral $\mathbf{A} \in \mathrm{RRA}$ with a flexible atom has a normal representation. Therefore, the universal-algebraic approach is applicable; in particular, we make heavy use of polymorphisms and their connection to primitive positive definability in later sections (cf. Theorem 2.45). Furthermore, we prove that every normal representation of a finite e relation algebra with a flexible atom has a Ramsey expansion (Section 4.2.2). Therefore, the tools from Section 2.7 can be applied, too. Finally we give some examples of relation algebras with a flexible atom (Section 4.2.3). We start with the definition of a flexible atom.

4.2 Definition Let $\mathbf{A} \in \mathrm{RA}$ and let $I := \{a \in A \mid a \leqslant \mathrm{Id}\}$. An atom $s \in A_0 \backslash I$ is called *flexible* if for all $a, b \in A \backslash I$ it holds that $s \leqslant a \circ b$.

This definition can for example be found in Chapter 11, Exercise 1 in the book [HH02]. Note that this definition does not require the relation algebra $\mathbf{A}$ to be integral. This is slightly more general than the definition by [Mad94,Mad06b]. As mentioned before, we show in the following section that it is sufficient for our result to classify the computational complexity of NSPs for finite representable relation algebras with a flexible atom that are additionally integral. This means that readers who prefer this second definition by [Mad94, Mad06b] (assuming integrality) can perfectly skip the following section and read the article with this other definition in mind. In this case relation algebras with a flexible atom are always implicitly integral.

4.2.1 Integral Relation Algebras

Let $\mathbf{A} \in \mathrm{RA}$ and let $I := \{a \in A \mid a \leqslant \mathrm{Id}\}$. The atoms in $I \cap A_0$ are called *identity atoms*. Therefore, $\mathbf{A}$ is integral if and only if $\mathbf{A}$ has exactly one identity atom.

4.3 Lemma *Let $\mathbf{A} \in \mathrm{RA}$ be finite. Then there exists for every atom s a unique $e_1 \in A_0$ with $0 < e_1 \leqslant \mathrm{Id}$ such that $s = e_1 \circ s$. Furthermore, if s is a flexible atom then for all $e_2 \in A_0$ with $0 < e_2 \leqslant \mathrm{Id}$ and $e_2 \neq e_1$ we have that $e_2 \circ \overline{\mathrm{Id}} = 0$.*

Proof: Note that $\mathrm{Id} \circ s = s$ by definition and therefore $e \circ s \subseteq s$ for all $e \in A_0$ with $0 < e \leqslant \mathrm{Id}$. Since s is an atom either $e \circ s = 0$ or $e \circ s = s$. By $\mathrm{Id} = \bigcup\{e \in A_0 \mid 0 < e \leqslant \mathrm{Id}\}$ and $\mathrm{Id} \circ s = s$ there exists at least one $0 < e \leqslant \mathrm{Id}$ with $e \circ s = s$. In the next step we prove uniqueness of such an element e. Assume for contradiction that there exist distinct $e_1, e_2 \in A_0$ with $0 < e_1 \leqslant \mathrm{Id}$ and $0 < e_2 \leqslant \mathrm{Id}$ such that $e_1 \circ s = s$ and $e_2 \circ s = s$. Note that $e_1 \circ e_2 = 0$ since e_1 and e_2 are identity atoms. Therefore, we get

$$0 = 0 \circ s = (e_1 \circ e_2) \circ s = e_1 \circ (e_2 \circ s) = e_1 \circ s = s,$$

which is a contradiction since s is an atom. This proves the first statement.

For the second statement assume for contradiction that there exists $e_2 \in A_0 \backslash \{e_1\}$ such that $e_2 \leqslant \mathrm{Id}$ and $e_2 \circ \overline{\mathrm{Id}} \neq 0$. Let a be an atom with $a \leqslant e_2 \circ \overline{\mathrm{Id}}$. Since $e_1 \circ e_2 = 0$ we get $e_1 \circ a \leqslant e_1 \circ (e_2 \circ \overline{\mathrm{Id}}) = (e_1 \circ e_2) \circ \overline{\mathrm{Id}} = 0 \circ \overline{\mathrm{Id}} = 0$. Since s is a flexible atom it holds that $s \leqslant a \circ a^{\smile}$ and therefore

$$s = e_1 \circ s \leqslant e_1 \circ (a \circ a^{\smile}) = (e_1 \circ a) \circ a^{\smile} = 0 \circ a^{\smile} = 0,$$

which is a contradiction. □

4.4 Proposition *Let* $\mathbf{A} \in \mathrm{RRA}$ *be finite and with a flexible atom. Then there exists a finite integral* $\mathbf{A}' \in \mathrm{RRA}$ *with a flexible atom such that the following statements hold:*

1. *There exists a polynomial-time Turing reduction from* $\mathrm{NSP}(\mathbf{A})$ *to* $\mathrm{NSP}(\mathbf{A}')$.
2. *There exists a polynomial-time many-one reduction from* $\mathrm{NSP}(\mathbf{A}')$ *to* $\mathrm{NSP}(\mathbf{A})$.
3. *The atom structure of* $\mathbf{A}$ *has a polymorphism that satisfies the Siggers identity if and only if the atom structure of* $\mathbf{A}'$ *has such a polymorphism (see Definition 4.12).*

Proof: If $\mathbf{A}$ is integral there is nothing to be shown. So assume that $\mathbf{A}$ is not integral and let s be a flexible atom. Let $\mathfrak{B}$ be a representation of $\mathbf{A}$ 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-sentences. Such a representation exists by Theorem 2.16. Let $(x, y) \in \overline{\mathrm{Id}}^{\mathfrak{B}}$ and let $e_1 \in A_0$ be the unique element with $e_1 \leqslant \mathrm{Id}$ and $s = e_1 \circ s$ that exists by Lemma 4.3. The second statement of Lemma 4.3 implies $e_1 \circ \overline{\mathrm{Id}} = \overline{\mathrm{Id}}$ and therefore we have that $(x, x) \in e_1^{\mathfrak{B}}$ and $(y, y) \in e_1^{\mathfrak{B}}$. Let $\mathfrak{C}'$ be the substructure of $\mathfrak{B}$ on the domain $\{x \in B \mid (x, x) \in e_1^{\mathfrak{B}}\}$. The set of relations of $\mathfrak{C}'$ clearly induces a proper relation algebra which is integral. We denote this representable relation algebra by $\mathbf{A}'$. Note that we can also consider A' as a subset of A. Let $\mathfrak{B}'$ be the representation of $\mathbf{A}'$ 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'-sentences. As before, such a representation exists by Theorem 2.16.

Proof of 1.: Note that if a connected instance of $\mathrm{CSP}(\mathfrak{B})$ is satisfiable, then either all variables are mapped to an atom from the subset of A_0 that corresponds to A'_0 or all variables are mapped to one element x with $e^*(x, x)$ and $e^* \in \{e \in A_0 \mid e \leqslant \mathrm{Id} \text{ and } e \neq e_1\}$. This leads to the following polynomial-time Turing reduction from $\mathrm{CSP}(\mathfrak{B})$ to $\mathrm{CSP}(\mathfrak{B}')$, which proves together with Theorem 2.16 the claim that there exists a polynomial-time Turing reduction from $\mathrm{NSP}(\mathbf{A})$ to $\mathrm{NSP}(\mathbf{A}')$. Consider the following algorithm: For a given primitive positive A-sentence φ, let $D(\varphi)$ be the canonical database of φ. The algorithm computes the conjunctive queries $\varphi_1, \ldots, \varphi_n$ of the connected components of $D(\varphi)$ (consider the proof of Proposition 2.18 for the terminology) . Then it defines new A-sentences φ'_i from φ_i for every $i \in \{1, \ldots, n\}$ by substituting every conjunct $a(x, y)$ with $a'(x, y)$ where $a' := a \backslash \{e \in A_0 \mid e \leqslant \mathrm{Id} \text{ and } e \neq e_1\}$. This resulting sentences are over the signature A'. Now the algorithm calls the decision procedure for $\mathrm{CSP}(\mathfrak{B}')$ on all inputs $\varphi'_1, \ldots, \varphi'_n$. Let $I \subseteq \{1, \ldots, n\}$ be the indices for which φ_i is not a satisfiable instance of $\mathrm{CSP}(\mathfrak{B}')$. In a last step the algorithm checks for every $i \in I$ whether there exists $e^* \in \{e \in A_0 \mid e \leqslant \mathrm{Id} \text{ and } e \neq e_1\}$ such that for every conjunct $a(x, y)$ of φ_i it holds that $e^* \leqslant a$. If this is true for every $i \in I$ the algorithm accepts the input, otherwise it rejects it.

Proof of 2.: Consider now an $\mathbf{A}'$-network $(V; f')$ and reduce this to the $\mathbf{A}$-network $(V; f')$. Suppose that $(V; f')$ is satisfiable in a representation $\mathfrak{D}'$ of $\mathbf{A}'$ by an assignment α. Let y_i be fresh elements for every atom $e_i \leqslant \mathrm{Id}$ with $e_i \neq e_1$. We build the disjoint union of $\mathfrak{D}'$ with one-element $\{e_i\}$-structures $(\{y_i\}; \{(y_i, y_i)\})$ and close the structure then under union and intersection of binary relations. This results in a representation of $\mathbf{A}$ that satisfies $(V; f')$ again by the assignment α. For the other direction, if $(V; f')$ is satisfiable in a representation $\mathfrak{D}$ of $\mathbf{A}$ we can again consider the substructure on the domain $(x, x) \in e_1^{\mathfrak{D}}$ and get a representation of $\mathbf{A}'$ that satisfies $(V; f')$.

Proof of 3.: Let g be a polymorphism of the atom structure of $\mathbf{A}$ that satisfies the Siggers identity. By assumption g satisfies

$$\forall x_1, \ldots, x_6 \in A_0.\, g(x_1, \ldots, x_6) \in \{x_1, \ldots x_6\}$$

and therefore the restriction of g to $(A_0 \cap A')^6$ is a polymorphism of the atom structure of $\mathbf{A}'$.

For the other direction choose an arbitrary ordering of the atoms $\{e \in A_0 \mid e \leqslant \mathrm{Id} \text{ and } e \neq e_1\} = \{l_1, \ldots, l_j\}$. If g is a Siggers polymorphism of the atom structure of $\mathbf{A}'$ one can extend g to an operation $g^* \colon A^6 \to A$ by defining

$$g^*(x_1, \ldots, x_6) := \begin{cases} \min(\{l_1, \ldots, l_j\} \cap \{x_1, \ldots, x_6\}) & \text{if } \{l_1, \ldots, l_j\} \cap \{x_1, \ldots, x_6\} \neq \emptyset, \\ g(x_1, \ldots, x_6) & \text{otherwise.} \end{cases}$$

It is easy to see that this operation satisfies also the Siggers identity. Furthermore, since every atom e from $\{e \in A_0 \mid e \leqslant \mathrm{Id} \text{ and } e \neq e_1\}$ is only contained in allowed triples of the from (e, e, e) it follows that f^* preserves the allowed triples from $\mathbf{A}$ (see after Definition 4.12). □

4.2.2 Normal Representations

Let $\mathbf{A} \in \mathrm{RRA}$ be for the rest of the section finite, integral, and with a flexible atom s. We consider the following subset of A:

$$A - s := \{a \in A \mid s \nleqslant a\}.$$

Let (V, g) be an $\mathbf{A}$-network and let $\mathfrak{C}$ be the corresponding A-structure (see paragraph before Definition 2.13). Let $\mathfrak{C} - s$ be the $(A - s)$-structure on the same domain V as $\mathfrak{C}$ such that for all $x, y \in V$ and $a \in (A - s)\setminus\{0\}$ we have

$$a^{\mathfrak{C}-s}(x, y) \text{ if and only if } (a^{\mathfrak{C}}(x, y) \vee (a \cup s)^{\mathfrak{C}}(x, y)).$$

We call $\mathfrak{C} - s$ the *s-free companion* of an **A**-network (V, f).

The next lemma follows directly from the definitions of flexible atoms and s-free companions.

4.5 Lemma *Let $\mathcal{C}$ be the class of s-free companions of atomic closed* **A***-networks. Then $\mathcal{C}$ has the free amalgamation property.*

Proof: Let $\mathfrak{A}$, $\mathfrak{B}_1$, and $\mathfrak{B}_2$ be structures in $\mathcal{C}$ with embeddings $e_1 \colon \mathfrak{A} \to \mathfrak{B}_1$ and $e_2 \colon \mathfrak{A} \to \mathfrak{B}_2$. Since s is a flexible atom and $\mathcal{C}$ is a class of s-free companions we get that the structure $\mathfrak{C} := \mathfrak{B}_1 \cup \mathfrak{B}_2$ is in $\mathcal{C}$. Therefore, the natural embeddings $f_1 \colon \mathfrak{B}_1 \to \mathfrak{C}$ and $f_2 \colon \mathfrak{B}_2 \to \mathfrak{C}$ prove the free amalgamation property of $\mathcal{C}$. □

As a consequence of this lemma we obtain the following.

4.6 Proposition **A** *has a normal representation $\mathfrak{B}$.*

Proof: Let $\mathcal{C}$ be the class from Lemma 4.5. This class is closed under taking substructures and isomorphisms. By Lemma 4.5 it also has the amalgamation property and therefore we get by Theorem 2.23 a homogeneous structure $\mathfrak{B}'$ with $\mathrm{Age}(\mathfrak{B}') = \mathcal{C}$. Let $\mathfrak{B}''$ be the expansion of $\mathfrak{B}'$ by the following relation

$$s(x, y) :\Leftrightarrow \bigwedge_{a \in A_0 \setminus \{s\}} \neg a^{\mathfrak{B}'}(x, y).$$

Let $\mathfrak{B}$ be the (homogeneous) expansion of $\mathfrak{B}''$ by all boolean combinations of relations from $\mathfrak{B}''$. Then $\mathfrak{B}$ is a representation of $\mathbf{A} \in \mathrm{RRA}$. Since $\mathrm{Age}(\mathfrak{B}')$ is the class of all atomic closed **A**-networks, $\mathfrak{B}$ is fully universal. The definition of s witnesses that $\mathfrak{B}$ is a square representation of **A**: for all elements $x, y \in B$ there exists an atom $a \in A_0$ such that $a^{\mathfrak{B}}(x, y)$ holds. □

The next theorem is another consequence of Lemma 4.5.

4.7 Theorem *Let $\mathfrak{B}$ be a normal representation of* **A**. *Let $\mathfrak{B}_<$ be the expansion of $\mathfrak{B}$ by a generic linear order. Then $\mathfrak{B}_<$ has the Ramsey property.*

Proof: Let $\mathfrak{B}'$ be the $(A_0 \setminus \{s\})$-reduct of $\mathfrak{B}$. The age of this structure has the free amalgamation property by Lemma 4.5. Therefore, Theorem 2.52 implies that the expansion of $\mathfrak{B}'$ by a generic linear order has the Ramsey property. By Remark 2.54 the structure $\mathfrak{B}_<$ also has the Ramsey property since $\mathfrak{B}_<$ and $(\mathfrak{B}')_<$ have the same automorphism group. □

4.8 Remark The binary first-order definable relations of $\mathfrak{B}_<$ form a proper relation algebra since $\mathfrak{B}_<$ has quantifier-elimination (see [Hod97]). By the definition of the generic order the atoms of this proper relation algebra are of the following form

$\circ$	Id	a	b
Id	Id	a	b
a	a	$\{\mathrm{Id}, a, b\}$	$\{a, b\}$
b	b	$\{a, b\}$	$\{\mathrm{Id}, a, b\}$

$\circ$	Id	a	b
Id	Id	a	b
a	a	$\{\mathrm{Id}, b\}$	$\{a, b\}$
b	b	$\{a, b\}$	$\{\mathrm{Id}, a, b\}$

Figure 4.1: Multiplication tables of relation algebras #18 (left) and #17 (right).

- $a^{\mathfrak{B}_<} \cap <^{\mathfrak{B}_<}$ for $a \in A_0 \backslash \{\mathrm{Id}\}$, or
- $a^{\mathfrak{B}_<} \cap >^{\mathfrak{B}_<}$ for $a \in A_0 \backslash \{\mathrm{Id}\}$, or
- Id,

where the relation $>^{\mathfrak{B}_<}$ consists of all tuples (x, y) such that $(y, x) \in <^{\mathfrak{B}_<}$ holds.

4.2.3 Examples

We give two concrete examples of finite integral symmetric relation algebras with a flexible atom (Examples 4.9 and 4.10), and a systematic way of building such algebras from arbitrary relation algebras (Example 4.11). The numbering of the algebras in the examples is from [AM94].

4.9 Example (Relation algebra #18) The representable relation algebra #18 has three atoms, namely the identity atom Id and two symmetric atoms a and b. The multiplication table for the atoms is given in Fig. 4.1. In this representable relation algebra the atoms a and b are flexible. Consider the countable, homogeneous, undirected graph $\mathfrak{R} = (V; E^{\mathfrak{R}})$, whose age is the class of all finite undirected graphs (see, e.g., [Hod97]), also called the *Random graph*. The expansion of $\mathfrak{R}$ by all binary first-order definable relations is a normal representation of the algebra #18. In this representation the atoms a and b are interpreted as the relation $E^{\mathfrak{R}}$ and the relation $N^{\mathfrak{R}}$, where $N^{\mathfrak{R}}$ is defined as $\neg E(x, y) \wedge x \neq y$.

4.10 Example (Relation algebra #17) The relation algebra #17 from Example 3.10 has three symmetric atoms. The multiplication table in Fig. 4.1 shows that in this algebra the element b is a flexible atom. To see that a is not a flexible atom, note that $a \nleq a \circ a = \{\mathrm{Id}, b\}$.

4.11 Example Consider an arbitrary finite, integral $\mathbf{A} = (A;\ \cup,\ ^{-},\ 0,\ 1,\ \mathrm{Id},\ ^{\smile},\ \circ)$. Clearly $\mathbf{A}$ does not have a flexible atom s in general. Nevertheless we can expand the domain of $\mathbf{A}$ to implement an "artificial" flexible atom.

Let s be some symbol not contained in A. Let us mention that every element in $\mathbf{A}$ can uniquely be written as a union of atoms from A_0. Let A' be the set of all subsets of $A_0 \cup \{s\}$. The set A' is the domain of our new algebra $\mathbf{A}'$. Note that on A' there exists the subset-ordering and A' is closed under set-union and complement (in $A_0 \cup \{s\}$) We define s to be symmetric and therefore get the following unary function * in $\mathbf{A}'$ as follows. For an element $x \in A'$ we define

$$x^* := \begin{cases} y^{\smile} \cup \{s\} & \text{if } x = y \cup \{s\} \text{ for } y \in A, \\ x^{\smile} & \text{otherwise.} \end{cases}$$

The new function symbol $\circ'_A$ in $\mathbf{A}'$ is defined on the atoms $A_0 \cup \{s\}$ as follows:

$$x \circ_{A'} y := \begin{cases} A_0 \cup \{s\} & \text{if } \{s\} = \{x, y\}, \\ (A_0 \backslash \{\mathrm{Id}\}) \cup \{s\} & \text{if } \{s, a\} = \{x, y\} \text{ for } a \in A_0 \backslash \{s, \mathrm{Id}\}, \\ \{a\} & \text{if } \{\mathrm{Id}, a\} = \{x, y\} \text{ for } a \in A_0 \cup \{s\}, \\ (x \circ y) \cup \{s\} & \text{otherwise.} \end{cases}$$

One can check that $\mathbf{A}' = (A';\ \cup,\ ^-,\ \varnothing,\ A_0 \cup \{s\},\ \mathrm{Id},\ ^*,\ \circ_{A'})$ is a finite integral representable relation algebra with a flexible atom s. Note that the forbidden triples of $\mathbf{A}'$ are exactly those of $\mathbf{A}$ together with triples which are permutations of (s, a, Id) for some $a \in A_0$.

4.3 Polynomial-time Tractability

In this section we introduce for every finite $\mathbf{A} \in \mathrm{RA}$ an associated finite structure, called the *atom structure* of $\mathbf{A}$. Note that it is closely related, but not the same, as the type structure introduced by [BM16]. In the context of relation algebras the atom structure has the advantage that its domain is the set of atoms of $\mathbf{A}$, rather than the set of 3-types, which would be the domain of the type structure of [BM16]; hence, our domain is smaller and has some advantages on which the main result of this section (Proposition 4.15) is based. Up to a some differences in the signature, our atom structure is the same as the atom structure introduced by [Lyn50] which was used there for different purposes see also [Mad82, HH01, HJK19].

Let $\mathfrak{B}$ be a normal representation of a finite $\mathbf{A} \in \mathrm{RRA}$. We will reduce $\mathrm{CSP}(\mathfrak{B})$ to the CSP of the atom structure of $\mathbf{A}$. This means that if the CSP of the atom structure is in P, then so are $\mathrm{CSP}(\mathfrak{B})$ and $\mathrm{NSP}(\mathbf{A})$. For our main result we will show later that every network satisfaction problem for a finite integral symmetric representable relation algebra with a flexible atom that cannot be solved in polynomial time by this method is NP-complete.

4.12 Definition 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 $x^{\mathfrak{A}_0} := \{a \in A_0 \mid a \leqslant x\}$,
- the binary relation $E^{\mathfrak{A}_0} := \{(a_1, a_2) \in A_0^2 \mid a_1^{\smile} = a_2\}$,
- the ternary relation $R^{\mathfrak{A}_0} := \{(a_1, a_2, a_3) \in A_0^3 \mid a_3 \leqslant a_1 \circ a_2\}$.

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 $R^{\mathfrak{A}_0}$.

4.13 Proposition *Let $\mathfrak{B}$ be a fully universal representation of a finite $\mathbf{A} \in \mathrm{RRA}$. There is a polynomial-time reduction from $\mathrm{CSP}(\mathfrak{B})$ to $\mathrm{CSP}(\mathfrak{A}_0)$.*

Proof: Let Ψ be an instance of $\mathrm{CSP}(\mathfrak{B})$ with variable set $X = \{x_1, \dots, x_n\}$. We construct an instance Φ of $\mathrm{CSP}(\mathfrak{A}_0)$ as follows. The variable set Y of Φ is given by $Y := \{(x_i, x_j) \in X^2 \mid i \leqslant j\}$. The constraints of Φ are of the two kinds:

1. Let $a \in A$ be an element of $\mathbf{A} \in \mathrm{RRA}$ and let $a((x_i, y_j))$ be an atomic formula of Ψ. If $i < j$, then we add the atomic (unary) formula $a((x_i, x_j))$ to Φ; otherwise we add the atomic formula $a^{\smile}((x_j, x_i))$. If $j = i$ we additionally add $\mathrm{Id}((x_i, x_j))$.
2. Let $x_i, x_j, x_l \in X$ be such that $i \leqslant j \leqslant l$. Then we add the atomic formula $R((x_i, x_j), (x_j, x_l), (x_i, x_l))$ to Φ.

It remains to show that this reduction is correct. Let $s\colon X \to B$ be a satisfying assignment for Ψ. This assignment maps every pair of variables x_i and x_j to a unique atom in A_0 and therefore induces a map $s'\colon Y \to A_0$. The map s' clearly satisfies all atomic formulas introduced by (1.). To see that it also satisfies all formulas introduced by (2.) note that s maps the elements $x_i, x_j, x_l \in X$ to a substructure of $\mathfrak{B}$, which does not induces a forbidden triple.

For the other direction assume that $s'\colon Y \to A_0$ is a satisfying assignment for Φ. This induces an A-structure $\mathfrak{X}$ on X (maybe with some identification of variables) as follows: we add (x_i, x_j) to the relation $a^{\mathfrak{X}}$ if $i \leqslant j$ and $s'((x_i, x_j)) = a$; if otherwise $j < i$ and $s'((x_j, x_i)) = a$ we add (x_i, x_j) to the relation $(a^{\smile})^{\mathfrak{X}}$. It is clear that no forbidden triple from $\mathbf{A}$ is induced by $\mathfrak{X}$. Also note that $\mathfrak{X}$ satisfies Ψ by the choice of the (unary) constraints of the first kind. Since $\mathfrak{B}$ is a fully universal representation the structure $\mathfrak{X}$ is a substructure of $\mathfrak{B}$. Hence, the instance Ψ is satisfiable in $\mathfrak{B}$. □

The atom structure has another property which is fundamental for our proof of Theorem 1. Recall that every canonical polymorphism f induces a behaviour $\overline{f}\colon A_0^n \to A_0$. In the next proposition we show that then $\overline{f}$ is a polymorphism of $\mathfrak{A}_0$. Moreover the other direction also holds. Every $g \in \mathrm{Pol}(\mathfrak{A}_0)$ is the behaviour of a canonical polymorphism of $\mathfrak{B}$.

4.14 Proposition *Let $\mathfrak{B}$ be a normal representation of a finite $\mathbf{A} \in \mathrm{RRA}$.*

1. *Let $g \in \mathrm{Pol}(\mathfrak{B})^{(n)}$ be canonical and let $\overline{g}\colon A_0^n \to A_0$ be its behaviour. Then $\overline{g} \in \mathrm{Pol}(\mathfrak{A}_0)^{(n)}$.*
2. *Let $f \in \mathrm{Pol}(\mathfrak{A}_0)^{(n)}$. Then there exists a canonical $g \in \mathrm{Pol}(\mathfrak{B})^{(n)}$ whose behaviour equals f.*

Proof: For (1): Let $g \in \mathrm{Pol}(\mathfrak{B})^{(n)}$ be canonical and let $c^1, \dots, c^n \in R^{\mathfrak{A}_0}$. Then by the definition of $R^{\mathfrak{A}_0}$ there exist tuples $x^1, \dots, x^n \in B^3$ such that for all $i \in \{1, \dots, n\}$ we have

$$c_1^{i\,\mathfrak{B}}(x_1^i, x_2^i),\ c_2^{i\,\mathfrak{B}}(x_2^i, x_3^i),\ \text{and } c_3^{i\,\mathfrak{B}}(x_1^i, x_3^i).$$

We apply the canonical polymorphism g and get $y := g(x^1, \dots, x^n) \in B^3$. Then there exists an allowed triple $(d_1, d_2, d_3) \in A_0^3$ such that

$$d_1^{\mathfrak{B}}(y_1, y_2),\ d_2^{\mathfrak{B}}(y_2, y_3),\ \text{and } d_3^{\mathfrak{B}}(y_1, y_3).$$

We have that $d = (d_1, d_2, d_3) \in R^{\mathfrak{A}_0}$ and by the definition of the behaviour of a canonical function we get $\overline{g}(c^1, \dots, c^n) = d$. The other relations in $\mathfrak{A}_0$ are preserved trivially and therefore $\overline{g} \in \mathrm{Pol}(\mathfrak{A}_0)^{(n)}$.

For (2): Since $\mathfrak{B}$ is fully universal and homogeneous it follows by a compactness argument see (e.g., Lemma 2 by [BD13]) that every countable A_0-structure which does not induce a forbidden triple and is square has a homomorphism to $\mathfrak{B}$. It is therefore enough to show that every operation $h\colon B^n \to B$ with behaviour f does not induce a forbidden triple in the image. Let $x^1, \ldots, x^n \in B^3$ be such that the application of a canonical function with behaviour f on $x^1, \ldots, x^n$ would give a tuple $y \in B^3$ with $d = (d_1, d_2, d_3) \in A_0^3$ such that

$$d_1^{\mathfrak{B}}(y_1, y_2),\ d_2^{\mathfrak{B}}(y_2, y_3),\ \text{and}\ d_3^{\mathfrak{B}}(y_1, y_3).$$

Since f preserves $R^{\mathfrak{A}_0}$ the triple d is not forbidden. □

Recall from Proposition 2.49 that polymorphisms of $\mathfrak{B}$ are edge-conservative. Note that this implies that polymorphisms of $\mathfrak{A}_0$ are conservative. In fact, Theorem 2.43 and the previous proposition imply the following.

4.15 Proposition *If* $\mathrm{Pol}(\mathfrak{B})$ *contains a canonical polymorphism s whose behaviour* $\bar{s}$ *is a Siggers operation in* $\mathrm{Pol}(\mathfrak{A}_0)$ *then* $\mathrm{CSP}(\mathfrak{B})$ *is in P.*

We demonstrate how this result can be used to prove polynomial-time tractability of $\mathrm{NSP}(\mathbf{A})$ for a symmetric, integral $\mathbf{A} \in \mathrm{RRA}$ with a flexible atom.

4.16 Example (Polynomial-time tractability the NSP of relation algebra #18)
The polynomial-time tractability of the NSP of the relation algebra #18 (see Example 4.9) was first shown by [CH04] (see also Section 8.4 of [BP15a]). Here we consider the following function $\bar{s}\colon \{\mathrm{Id}, a, b\}^6 \to \{\mathrm{Id}, a, b\}$.

$$\bar{s}(x_1, \ldots, x_6) := \begin{cases} a & \text{if } a \in \{x_1, \ldots, x_6\}, \\ b & \text{if } b \in \{x_1, \ldots, x_6\} \text{ and } a \notin \{x_1, \ldots, x_6\}, \\ \mathrm{Id} & \text{otherwise.} \end{cases}$$

Let $\mathfrak{N}'$ be the normal representation of the algebra #18 given in Example 4.9. Note that $\bar{s}$ is the behaviour of an injective, canonical polymorphism of $\mathfrak{N}$. The injectivity follows from the last line of the definition; if $\bar{s}(x_1, \ldots, x_6) = \mathrm{Id}$ then $\{x_1, \ldots, x_6\} = \{\mathrm{Id}\}$. Therefore $\bar{s}$ preserves all allowed triples, since in the algebra #18 the only forbidden triples involve Id. One can check that $\bar{s}$ is a Siggers operation and therefore we get by Proposition 4.15 that NSP(#18) is in P.

4.17 Example Consider the construction of relation algebras with a flexible atom from Example 4.11. It is easy to see that $\mathrm{NSP}(\mathbf{A})$ for a finite integral $\mathbf{A} \in \mathrm{RRA}$ has a polynomial-time reduction to $\mathrm{NSP}(\mathbf{A}')$ where $\mathbf{A}'$ is the relation algebra with a flexible atom that is constructed in Example 4.11. We get as a consequence that if a normal representation of $\mathbf{A}'$ satisfies the condition of Proposition 4.15 then $\mathrm{NSP}(\mathbf{A})$ is in P.

Additionally to Proposition 4.15 we get by Theorem 2.43 another important observation.

4.18 Corollary *If* $\mathrm{Pol}(\mathfrak{A}_0)$ *does not have a Siggers operation then there exist elements* $a_1, a_2 \in A_0$ *such that the restriction of every operation from* $\mathrm{Pol}(\mathfrak{A}_0)^{(n)}$ *to* $\{a_1, a_2\}^n$ *is a projection.*

4.4 Network Consistency Problems

The purpose of this section is to give an additional perspective on the class of network satisfaction problems of finite symmetric integral representable relation algebras with a flexible atom. Even more, we define these computational problems in this section completely without the use of the relation algebra framework. Our classification result for these problems does not depend on the content of this section and the reader may skip it.

We introduce a class of computational decision problems which we call *network consistency problems* (*NCPs*). It is easy to see that NCPs are in a 1-to-1 correspondence with NSPs of finite, symmetric, integral $\mathbf{A} \in \mathrm{RRA}$ with a flexible atom.

4.19 Definition Let A be a finite set and $R \subseteq A^3$. Then R is called *totally symmetric* if for all bijections $\pi\colon \{1,2,3\} \to \{1,2,3\}$ we have

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

We call an element $p \in A$ *identity element* if for all $x, y \in A$ the following holds:

$$(p, x, y) \in R \Leftrightarrow x = y.$$

A structure $(A; R)$ is called a *stencil* if R is totally symmetric and it contains an identity element.

4.20 Definition Let $(G; F)$ be an undirected graph and let Q be a set. We call a map $c\colon F \to Q$ an *edge Q-coloring of* $(G; F)$ if for all $x, y \in G$ with $(x, y) \in F$ it holds that $c((x, y)) = c((y, x))$.

For each fixed stencil, we define an NCP as follows.

4.21 Definition Let (A, R) be a stencil. The *network completion problem of* (A, R), denoted by $\mathrm{NCP}(A, R)$, is the following problem. Given a finite undirected graph $(G; F)$ with an edge $\mathcal{P}(A)$-coloring f the task is to decide whether there exists an edge A-coloring f' of $(G; F)$ such that

1. for all $x, y \in G$ with $(x, y) \in F$ it holds that $f'((x, y)) \in f((x, y))$.
2. for all $x, y, z \in G$ with $(x, y), (y, z), (x, z) \in F$ we have

$$(f'((x, y)), f'((y, z)), f'((x, z))) \in R.$$

The following proposition illustrates how NCPs correspond to a certain class of NSPs.

4.22 Proposition *The class of NCPs and the class of NSPs for finite symmetric integral representable relation algebras with a flexible atom are in a natural 1-to-1 correspondence such that corresponding problems are polynomial-time equivalent.*

Proof: Let (A', R') be a stencil with $p \in A'$ according to (2) in Definition 4.19 and let s be new element with $s \notin A'$. We define a relational structure $\mathfrak{D}$ as follows. The domain of $\mathfrak{D}$ is the set $A' \cup \{s\}$. We assume that $\mathfrak{D}$ has every subset of its domain as a unary relation. Furthermore $\mathfrak{D}$ contains the binary relation $E^{\mathfrak{D}} := \{(x, x) \mid x \in A' \cup \{s\}\}$ and the ternary relation $R^{\mathfrak{D}}$ which is defined as follows:

$$\begin{aligned}(x, y, z) \in R^{\mathfrak{D}} :\Leftrightarrow \big(&(x, y, z) \in R' \\ &\vee (s \in \{x, y, z\} \wedge p \notin \{x, y, z\}) \\ &\vee ((x, y, z) \in \{(p, s, s), (s, p, s), (s, s, p)\})\big).\end{aligned}$$

It is easy to see that one can find a finite symmetric integral algebra $\mathbf{A} \in \mathrm{RRA}$ with domain $\mathcal{P}(A' \cup \{s\})$ and a flexible atom s such that $\mathfrak{D}$ is the atom structure of $\mathbf{A}$. Furthermore, given a finite symmetric integral algebra $\mathbf{A} \in \mathrm{RRA}$ with a flexible atom s let $\mathfrak{D}$ be the atom structure of $\mathbf{A}$. Let $(A'; R')$ be the substructure of the $\{R\}$-reduct of $\mathfrak{D}$ induced by $D \backslash \{s\}$. By the properties of $\mathfrak{D}$ we get that $(A'; R')$ is a stencil.

We show that the instances of $\mathrm{NCP}(A', R')$ and $\mathrm{NSP}(\mathbf{A})$ are in a natural 1-to-1 correspondence that preserves the acceptance condition of the computational problems. Let $(G; F)$ be a finite undirected graph with an edge $\mathcal{P}(A')$-coloring f. We define an $\mathbf{A}$-network $(G; g)$ by defining

$$g(x, y) = \begin{cases} f(x, y) & (x, y) \in F, \\ \{p\} & (x, y) \notin F \text{ and } x = y \\ \{s\} & \text{else.} \end{cases}$$

It is easy to see that $(G; F)$ is an accepted instance of $\mathrm{NCP}(A', R')$ if and only if $(G; g)$ is an accepted instance of $\mathrm{NSP}(\mathbf{A})$. Since we can reverse this and find for every $\mathbf{A}$-network $(G; g)$ a finite undirected graph $(G; F)$ with an edge $\mathcal{P}(A')$-coloring f such that each of them is an accepted instance if and only if the other one is. These two reductions show that the computational decision problems $\mathrm{NCP}(A', R')$ and $\mathrm{NSP}(\mathbf{A})$ are polynomial-time equivalent. □

We end this section by providing a rich source of examples for NCPs.

4.23 Example ("Distance problems") Let $A \subset \mathbb{Q}$ be an arbitrary finite set that contains 0. We define the relation $R \subseteq A^3$ of all tuples which satisfy all instantiations of the triangle inequality, i.e.

$$(a_1, a_2, a_3) \in R \ :\Leftrightarrow \ (a_1 \leqslant a_2 + a_3) \ \wedge \ (a_2 \leqslant a_1 + a_3) \ \wedge \ (a_3 \leqslant a_1 + a_2),$$

where the addition is meant to be the usual one on rational numbers. The relation R is by definition totally symmetric and the element 0 is an identity element. Therefore, $(A; R)$ is a stencil.

Now consider a finite undirected graph $(G; F)$ with an edge $\mathcal{P}(A)$-coloring f. This can be seen as a labeling of each edge in the graph by a set of possible (or allowed) distances. The computational task of $\mathrm{NCP}(A, R)$ is to decide whether one can choose for each edge (x, y) one of the possible distance such that in the end this choice satisfies on each triangle of edges the triangle inequalities of metric spaces.

By Proposition 4.22 there exists a finite symmetric integral algebra $\mathbf{A}' \in \mathrm{RRA}$ with a flexible atom s such that $\mathrm{NCP}(A, R)$ and $\mathrm{NSP}(\mathbf{A}')$ are polynomial-time equivalent. By the proof of Proposition 4.22 we have that the domain of $\mathbf{A}'$ is equal to $\mathcal{P}(A \cup \{s\})$. It is easy to observe that $A \cup \{s\}$ is the set of atoms A_0' from $\mathbf{A}'$.

We define an operation $f\colon {A_0'}^6 \to A_0'$ as follows:

$$f(x_1, \dots, x_6) = \begin{cases} s & s \in \{x_1, \dots, x_6\}, \\ \max\{x_1, \dots, x_6\} & \text{otherwise}, \end{cases}$$

where the max operation is the usual from in $\mathbb{Q}$.

The allowed triples of $\mathbf{A}'$ are, up to triples that involve the flexible atom s, those which arise from valid triangle inequalities. For this reason the operation f preserves the allowed triples of $\mathbf{A}'$. Moreover, one can check that f satisfies the Siggers identity. This implies that all these "distance problems" satisfy the first condition in Theorem 4.1 and are therefore solvable by a polynomial-time algorithm. Furthermore, let $\mathbf{A}$ be the algebra that arises from $\mathbf{A}'$ by deleting the flexible atom s. If $\mathbf{A}$ is a representable relation algebra with a normal representation we get by the argument from Example 4.17 that $\mathrm{NSP}(\mathbf{A})$ is also in P.

It follows from Lemma 3.1.1 in [PR96] (see also Proposition 2.7.4 in [Con15]) that if A is a finite initial segment of integers, then $\mathbf{A}$ has a normal representation and therefore $\mathrm{NSP}(\mathbf{A})$ is in P.

4.5 Binary Injective Polymorphisms

We give in this section a proof of the following proposition.

4.24 Proposition *Let $\mathfrak{B}$ be a normal representation of a finite, symmetric, integral* $\mathbf{A} \in \mathrm{RRA}$ *with a flexible atom s. If* $\mathrm{HSP}^{\mathrm{fin}}(\{\mathrm{Pol}(\mathfrak{B})\})$ *does not contain a 2-element algebra where all operations are projections, then $\mathfrak{B}$ has a binary injective polymorphism.*

This statement is a consequence of well known results that can be found in the book by [Bod21] and results by [MP22] applied to the class of normal representations of finite, symmetric, integral $\mathbf{A} \in \mathrm{RRA}$ with a flexible atom s. An operation $f \in \mathrm{Pol}(\mathfrak{B})$ is called *essentially unary* if it depends on at most one of its variables and f is called *essential* otherwise.

Following the terminology of [MP22], we now define *free 2-orbits*. The existence of a free 2-orbit appeared under the name *'orbital extension property'* for example in the habilitation thesis of [Bod12].

4.25 Definition Let $\mathfrak{B}$ be a structure. A 2-orbit O of $\mathrm{Aut}(\mathfrak{B})$ is called *free* if for all elements $x, y \in B$ there exists $z \in B$ with $(z, x) \in O$ and $(z, y) \in O$.

Note that if $\mathrm{Aut}(\mathfrak{B})$ has a free 2-orbit then it is transitive. The following theorem generalises a fact that was first proved for first-order reducts of $(\mathbb{Q}; <)$ by [BK09b].

4.26 Proposition (Lemma 5.3.10 in [Bod12]) *Let $\mathfrak{B}$ be a structure such that $\mathfrak{B}$ has a free 2-orbit. If* $\mathrm{Pol}(\mathfrak{B})$ *contains an essential operation then it contains a binary essential operation.*

The following is essentially taken from the article by [MP22].

4.27 Definition Let $\mathfrak{B}$ be a structure. Then the *canonical binary structure* of $\mathfrak{B}$ is the structure with domain B and a binary relation for each 2-orbit O of $\mathrm{Aut}(\mathfrak{B})$ such that $(x, y) \in O$ implies $x \neq y$.

4.28 Definition A τ-structure $\mathfrak{B}$ has *finite duality* if there exists a finite set $\mathcal{F}$ of finite τ-structures such that a τ-structure $\mathfrak{I}$ has a homomorphism to $\mathfrak{B}$ if and only if no element of $\mathcal{F}$ has a homomorphism to $\mathfrak{I}$.

We establish finite duality for the class of structures which is important for our classification purposes.

4.29 Lemma *Let $\mathfrak{B}$ be a normal representation of a finite, integral* $\mathbf{A} \in \mathrm{RRA}$ *with a flexible atom s. Then the canonical binary structure of $\mathfrak{B}$ has finite duality.*

Proof: Let $\tau := A_0 \backslash \{\mathrm{Id}\}$. Note that since $\mathbf{A}$ is integral and $\mathfrak{B}$ is homogeneous, the canonical binary structure $\mathfrak{C}$ of $\mathrm{Aut}(\mathfrak{B})$ is precisely the τ-reduct of $\mathfrak{B}$. Let $\mathcal{F}$ be the set of all τ-structures with domain $\{1, 2, 3\}$ that do not have a homomorphism to $\mathfrak{C}$. We show that the set $\mathcal{F}$ witnesses the finite duality of the canonical binary structure $\mathfrak{C}$ of $\mathfrak{B}$. Let $\mathfrak{I}$ be a τ-structure with a homomorphism to $\mathfrak{C}$. If there exists $\mathfrak{F} \in \mathcal{F}$ with a homomorphism to $\mathfrak{I}$, then $\mathfrak{F}$ also has a homomorphism to $\mathfrak{C}$, contradicting the choice of $\mathcal{F}$. For the other direction assume that no element from $\mathcal{F}$ has a homomorphism to $\mathfrak{I}$. Let $\mathfrak{I}'$ be the τ-expansion of the $(\tau \backslash \{s\})$-reduct of $\mathfrak{I}$ where the relation $s^{\mathfrak{I}'}$ is defined by

$$s^{\mathfrak{I}'} := \{(x,y) \in I^2 \mid (x,y) \in s^{\mathfrak{I}} \vee (x \neq y \wedge \forall a \in \tau \backslash \{s\}.\, (x,y) \notin a^{\mathfrak{I}})\}.$$

By the definition of the flexible atom s it follows that no element from $\mathcal{F}$ has a homomorphism to $\mathfrak{I}'$. This implies that for all distinct elements x, y from $\mathfrak{I}'$ the tuple (x,y) is in at most one relation from τ. The definition of $\mathfrak{I}'$ ensures that (x,y) is in at least one relation from τ. Recall that Proposition 2.21 states that $\mathfrak{C}$ is finitely bounded by τ-structures of size at most three. Assume that one of those bounds $\mathfrak{N}$ embeds into $\mathfrak{I}'$. This implies by what we noted before that all elements of $\mathfrak{N}$ are in precisely one relation from τ. On the other hand $\mathfrak{N}$ is not in $\mathcal{F}$ and therefore has a homomorphism to $\mathfrak{C}$. Since all elements of $\mathfrak{N}$ are related by precisely one relation from τ, this homomorphism needs to be an embedding, contradicting our assumption on $\mathfrak{N}$ to be a bound. Therefore, none of the bounds of $\mathfrak{C}$ embeds into $\mathfrak{I}'$, which means that $\mathfrak{I}'$ is a substructure of $\mathfrak{C}$. Clearly, there exists a homomorphism from $\mathfrak{I}$ to $\mathfrak{I}'$ which proves the lemma. □

The following proposition about the existence of injective operations is from [MP22], building on ideas of [BP15a] and [BMPP19].

4.30 Proposition ([MP22]) *Let $\mathfrak{B}$ be a homogeneous structure such that $\mathrm{Aut}(\mathfrak{B})$ is transitive and such that the canonical binary structure of $\mathrm{Aut}(\mathfrak{B})$ has finite duality. If $\mathrm{Pol}(\mathfrak{B})$ contains a binary essential operation that preserves $\neq$ then it contains a binary injective operation.*

We are now able to prove the main result of this section.

Proof of Proposition 4.24: Note that since $\mathbf{A}$ is integral and $\mathfrak{B}$ is homogeneous, the flexible atom s is a free 2-orbit of $\mathrm{Aut}(\mathfrak{B})$. Furthermore, $\mathrm{Aut}(\mathfrak{B})$ is transitive. Suppose that $\mathrm{HSP}^{\mathrm{fin}}(\{\mathrm{Pol}(\mathfrak{B})\})$ does not contain a 2-element algebra where all operations are projections. Since all operations of $\mathrm{Pol}(\mathfrak{B})$ are edge conservative, it follows that $\mathrm{Pol}(\mathfrak{B})$ contains an operation that does not behave as a projection on $\{s, \mathrm{Id}\}$. This implies that $\mathrm{Pol}(\mathfrak{B})$ contains an essential operation. By Proposition 4.26, $\mathrm{Pol}(\mathfrak{B})$ must also contain a binary essential operation. Since the canonical binary structure of $\mathfrak{B}$ has finite duality by Lemma 4.29 we can apply Proposition 4.30 and get that $\mathrm{Pol}(\mathfrak{B})$ contains a binary injective operation. □

The following shows how to use Proposition 4.24 to reprove the hardness result for a concrete $\mathbf{A} \in \mathrm{RRA}$ from Example 3.10 and Example 4.10.

4.31 Example (Hardness of relation algebra #17, see Example 3.10)
Let $\mathfrak{B}$ be the normal representation of the relation algebra #17 mentioned in Example

3.10. We claim that the structure $\mathfrak{B}$ does not have a binary injective polymorphism. To see this, consider a substructure of $\mathfrak{B}^2$ on elements $x, y, z \in V^2$ such that $(E, =)(x, y)$, $(=, E)(y, x)$, and $(E, E)(x, z)$. Assume $\mathfrak{B}$ has a binary injective polymorphism f. This means that $\overline{f}(E, \mathrm{Id}) = E = \overline{f}(\mathrm{Id}, E)$ holds. Then we get that $E(f(x), f(y))$, $E(f(y), f(z))$, and $E(f(x), f(z)$ hold in $\mathfrak{B}$, which is a contradiction, since in $\mathfrak{B}$ triangles of this form are forbidden. By the contraposition of Proposition 4.24 it follows that $\mathrm{HSP}^{\mathrm{fin}}(\{\mathrm{Pol}(\mathfrak{B})\})$ contains a 2-element algebra where all operations are projections. We conclude with Theorem 2.47 that $\mathrm{NSP}(\#17)$ is an NP-hard problem.

4.6 From Partial to Total Canonical Behaviour

In this section we prove that in many cases the existence of a polymorphism with a certain partial behaviour implies the existence of a canonical polymorphism with the same partial behaviour. Following this idea we start in Section 4.6.1 with the proof that the existence of an injective polymorphism implies the existence of a canonical injective polymorphism. In some cases the existence of an $\{a, b\}$-canonical polymorphism implies the existence of a canonical polymorphism with the same behaviour on $\{a, b\}$. We prove this separately for binary (Section 4.6.2) and ternary (Section 4.6.3) operations, making use of the binary injective polymorphism that exists by the results from Section 4.5 and Section 4.6.1.

Let us remark that most proofs of this section would fail if the representable relation algebra $\mathbf{A}$ was not symmetric. Indeed, every representable relation algebra that contains a non-symmetric atom a normal representation would not satisfy Proposition 4.33 below, since the stated behaviour is not well-defined.

We assume for this section that $\mathfrak{B}$ is a normal representation of a finite, symmetric, integral $\mathbf{A} \in \mathrm{RRA}$ with a flexible atom s. Let furthermore $\mathfrak{B}_<$ be the expansion of $\mathfrak{B}$ by the generic linear order. The structure $\mathfrak{B}_<$ exists by the observations in Section 4.2.2.

4.32 Proposition *Let $f \in \mathrm{Pol}(\mathfrak{B})^{(n)}$ be injective. Then there exists a polymorphism $f_<$ of $\mathfrak{B}_<$ and an injective endomorphism e of $\mathfrak{B}$ such that*

$$f = e \circ f_<$$

as mappings from B^n to B.

Proof: Let $U := f(B^n)$ and consider the substructure $\mathfrak{U}$ induced by $\mathfrak{B}$ on U. There exists a linear ordering on B^n, namely the lexicographic order given by the linear order of $\mathfrak{B}_<$ on each coordinate.

Let $\mathfrak{U}_<$ be the expansion of $\mathfrak{U}$ by the linear order that is induced by the lexicographic linear order of $\mathfrak{B}_<$ on the preimage. This is well defined since f is injective. By the definition of $\mathfrak{B}_<$ and a compactness argument the structure $\mathfrak{U}_<$ embeds into $\mathfrak{B}_<$. In this way we obtain a homomorphism $f_<$ from $\mathfrak{B}_<^n$ to $\mathfrak{B}_<$. Again by a compactness argument also an endomorphism e with the desired properties exists. □

4.6.1 Canonical Binary Injective Polymorphisms

We prove in this section that the existence of an injective polymorphism implies the existence of a canonical injective polymorphism. We say that a polymorphism f of $\mathfrak{B}_<$ is *canonical with respect to* $\mathfrak{B}_<$ if f satisfies Definition 2.50, where the underlying representable relation algebra is the proper relation algebra induced by the binary first-order definable relations (i.e., unions of 2-orbits) of $\mathfrak{B}_<$. Note that in a normal representation the set of 2-orbits equals the set of the interpretations of the atoms of the representable relation algebra.

4.33 Proposition *Let f be a binary polymorphism of $\mathfrak{B}_<$ that is canonical with respect to $\mathfrak{B}_<$. Let $h\colon A_0^2 \to A_0$ be the map such that for all $x, y, z \in A_0$*

$$h(x,y) = z \Leftrightarrow \overline{f}(x^{\mathfrak{B}_<} \cap \leqslant^{\mathfrak{B}_<}, y^{\mathfrak{B}_<} \cap \leqslant^{\mathfrak{B}_<}) = z^{\mathfrak{B}_<} \cap \leqslant^{\mathfrak{B}_<},$$

(*cf. Remark 4.8*). *Then h is well defined and there exists a canonical binary polymorphism of $\mathfrak{B}$ with behaviour h.*

Proof: The function h is well defined since all atoms are symmetric. We show that there exists a canonical polymorphism of $\mathfrak{B}$ that has h as a behaviour. Consider the following structure $\mathfrak{A}$ on the domain B^2. Let $x, y \in B^2$ and let $a, a_1, a_2 \in A_0$ be atoms of $\mathbf{A}$ with $a_1^{\mathfrak{B}}(x_1, y_1)$ and $a_2^{\mathfrak{B}}(x_2, y_2)$. Then we define that $a^{\mathfrak{A}}(x, y)$ holds if and only if $h(a_1, a_2) = a$.

We show in the following that $\mathfrak{A}$ has a homomorphism to $\mathfrak{B}$. This is enough to prove the statement, because a homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ is a canonical polymorphism of $\mathfrak{B}$. Since $\mathfrak{B}$ is homogeneous is suffices to show that every finite substructure of $\mathfrak{A}$ homomorphically maps to $\mathfrak{B}$. Let $\mathfrak{F}$ be a finite substructure of $\mathfrak{A}$ and assume for contradiction that $\mathfrak{F}$ does not homomorphically map to $\mathfrak{B}$. We can view $\mathfrak{F}$ as an atomic $\mathbf{A}$-network. Since $\mathfrak{B}$ is fully universal $\mathfrak{F}$ is not closed. There must exist elements $b^1, b^2, b^3 \in B^2$ of $\mathfrak{F}$ and atoms $a_1, a_2, a_3 \in A_0$ such that $a_1 \not\leqslant a_2 \circ a_3$ holds in $\mathbf{A}$ and

$$a_1^{\mathfrak{F}}(b^1, b^3),\ a_2^{\mathfrak{F}}(b^1, b^2),\ \text{and } a_3^{\mathfrak{F}}(b^2, b^3).$$

This means that the substructure induced on the elements b^1, b^2, b^3 by $\mathfrak{F}$ contains a forbidden triple.

Now we consider the substructures that are induced on b_1^1, b_1^2, b_1^3 and b_2^1, b_2^2, b_2^3 by $\mathfrak{B}$. Our goal is to order these elements such that for all $i, j \in \{1, 2, 3\}$

$$\neg(b_1^i < b_1^j \wedge b_2^i > b_2^j). \tag{4.1}$$

If we achieve this we know that there exist elements in $\mathfrak{B}_<$ that induce isomorphic copies of the induced structures of the elements b_1^1, b_1^2, b_1^3 and b_2^1, b_2^2, b_2^3 with the additional ordering. Now the application of the polymorphism f on these elements results in a structure whose A_0-reduct is isomorphic to the substructure induced by b^1, b^2 and b^3 on $\mathfrak{F}$ by the definition of the canonical behaviour h. This contradicts our assumption because a polymorphism can not have a forbidden substructure in its image.

It remains to show that we can choose orderings on the elements b_1^1, b_1^2, b_1^3 and b_2^1, b_2^2, b_2^3 such that (4.1) holds. Without loss of generality we can assume that

$$\{b_1^1, b_1^2, b_1^3\} \cap \{b_2^1, b_2^2, b_2^3\} = \varnothing$$

holds. Now consider the following cases:

1. $|\{b_1^1, b_1^2, b_1^3\}| = 3$ and $|\{b_2^1, b_2^2, b_2^3\}| = 3$.

   We can obviously choose linear orders on both sets such that (4.1) holds.

2. $|\{b_1^1, b_1^2, b_1^3\}| = 2$ and $|\{b_2^1, b_2^2, b_2^3\}| = 3$.
   Assume that $\mathrm{Id}^{\mathfrak{B}}(b_1^1, b_1^2)$ holds then the possible orders are
$$b_1^1 = b_1^2 < b_1^3 \text{ and } b_2^1 < b_2^2 < b_2^3.$$
3. $|\{b_1^1, b_1^2, b_1^3\}| = 2$ and $|\{b_2^1, b_2^2, b_2^3\}| = 2$.
   First consider the case that $\mathrm{Id}^{\mathfrak{B}}(b_1^1, b_1^2)$ and $\mathrm{Id}^{\mathfrak{B}}(b_2^1, b_2^2)$ hold. Then we choose as orders
$$b_1^1 = b_1^2 < b_1^3 \text{ and } b_2^1 = b_2^2 < b_2^3.$$
   In the second possible case we can assume without loss of generality that $\mathrm{Id}^{\mathfrak{B}}(b_1^1, b_1^2)$ and $\mathrm{Id}^{\mathfrak{B}}(b_2^2, b_2^3)$ hold. Note that otherwise we could change the role of two of the tuples b^1, b^2 and b^3 and get this case. The compatible order is then
$$b_1^1 = b_1^2 < b_1^3 \text{ and } b_2^1 < b_2^2 = b_2^3.$$
4. $|\{b_1^1, b_1^2, b_1^3\}| = 1$ and $|\{b_2^1, b_2^2, b_2^3\}| = 3$.
   In this case we choose the order
$$b_1^1 = b_1^2 = b_1^3 \text{ and } b_2^1 < b_2^2 < b_2^3.$$
5. $|\{b_1^1, b_1^2, b_1^3\}| = 1$ and $|\{b_2^1, b_2^2, b_2^3\}| = 2$.
   Assume that $\mathrm{Id}^{\mathfrak{B}}(b_2^1, b_2^2)$ holds and that we have
$$b_1^1 = b_1^2 = b_1^3 \text{ and } b_2^1 = b_2^2 < b_2^3.$$
6. $|\{b_1^1, b_1^2, b_1^3\}| = 1$ and $|\{b_2^1, b_2^2, b_2^3\}| = 1$.
   For this case we trivially get
$$b_1^1 = b_1^2 = b_1^3 \text{ and } b_2^1 = b_2^2 = b_2^3.$$

Note that up to the symmetry of the arguments for both coordinates these are all the possible cases. This completes the proof of the proposition. □

4.34 Corollary *Suppose that $\mathfrak{B}$ has a binary injective polymorphism. Then $\mathfrak{B}$ also has a canonical binary injective polymorphism.*

Proof: By Proposition 4.32 we may assume that there exists also an injective polymorphism of $\mathfrak{B}_<$. The structure $\mathfrak{B}_<$ has the Ramsey property by Theorem 4.7. Therefore, Theorem 2.53 implies that there also exists an injective canonical polymorphism g of $\mathfrak{B}_<$. According to Proposition 4.33 the restriction of the behaviour $\overline{g}$ to the 2-orbits that satisfy $x \leqslant y$ induces the behaviour of a canonical polymorphism of $\mathfrak{B}$ which is also injective. □

4.6.2 Canonical $\{a, b\}$-symmetric Polymorphisms

We will now use the results about binary injective polymorphism from Section 4.6.1 to show the existence of a canonical $\{a, b\}$-symmetric polymorphism in case there exists an $\{a, b\}$-symmetric polymorphism.

4.35 Lemma *Let $a, b \in A_0 \backslash \{\mathrm{Id}\}$ be atoms. Then every binary $\{a, b\}$-symmetric polymorphism of $\mathfrak{B}$ is injective.*

Proof: Let f be an $\{a, b\}$-symmetric polymorphism. Without loss of generality $\overline{f}(a, b) = a = \overline{f}(b, a)$. Assume for contradiction that f is not injective. This means that there exist $c \in A_0$ and $x, y \in B^2$ with $(c, \mathrm{Id})(x, y)$ (for the notation see Definition 2.48) such that $\mathrm{Id}(f(x), f(y))$ holds.

Case 1: $s \notin \{a, b\}$. Since s is a flexible atom we may choose $z \in B^2$ such that $(a, b)(z, x)$ and $(s, b)(z, y)$ hold. By the choice of the polymorphism f we get $a(f(z), f(x))$ and $(s \cup b)(f(z), f(y))$ which induces either the forbidden triple (Id, s, a) or the forbidden triple (Id, b, a) on $f(x), f(y)$, and $f(z)$.

Case 2: $s = a$. We choose $z \in B^2$ such that $(a, b)(z, x)$ and $(b, b)(z, y)$. This is possible since a is the flexible atom. We obtain $a(f(z), f(x))$ and $b(f(z), f(y))$ which again induces a forbidden triple on $f(x), f(y)$, and $f(z)$.

Case 3: $s = b$. We choose $z \in B^2$ such that $(a, b)(z, x)$ and $(b, b)(z, y)$. This is possible since a is the flexible atom. We obtain $a(f(z), f(x))$ and $b(f(z), f(y))$ which again induces a forbidden triple on $f(x), f(y)$, and $f(z)$.

Since we obtained in all cases a contradiction we conclude that f is injective. □

4.36 Proposition *Let $a, b \in A_0 \backslash \{\mathrm{Id}\}$ be atoms. If $\mathfrak{B}$ has a binary $\{a, b\}$-symmetric polymorphism, then $\mathfrak{B}$ has also a binary canonical $\{a, b\}$-symmetric polymorphism.*

Proof: Let f be the binary $\{a, b\}$-symmetric polymorphism. By Lemma 4.35 we know that f is injective. By Proposition 4.32 it induces a polymorphism $f_<$ on $\mathfrak{B}_<$. The structure $\mathfrak{B}_<$ has the Ramsey property by Theorem 4.7. Let g be the canonization of $f_<$ that exists by Theorem 2.53. The restriction of the behaviour $\overline{g}$ to the 2-orbits that satisfy $x \leqslant y$ induces by Proposition 4.33 the behaviour of a canonical polymorphism h of $\mathfrak{B}$. The way we obtained h ensures that h is $\{a, b\}$-symmetric with the same behaviour on $\{a, b\}$ as f. □

The following is an easy observation about $\{a, \mathrm{Id}\}$-symmetric polymorphisms that we will use several times.

4.37 Observation *Let $a \nleqslant \mathrm{Id}$ be an atom and f an $\{a, \mathrm{Id}\}$-symmetric polymorphism of $\mathfrak{B}$. Then $\overline{f}(a, \mathrm{Id}) = a = \overline{f}(\mathrm{Id}, a)$.*

Proof: Suppose for contradiction that $\overline{f}(a, \mathrm{Id}) = \mathrm{Id} = \overline{f}(\mathrm{Id}, a)$. Let $x, y, z \in B^2$ be such that

$$(a, \mathrm{Id})(x, y), \ (\mathrm{Id}, a)(y, z), \ \text{and} \ (a, a)(x, z)$$

and consider the substructure of $\mathfrak{B}$ that is induced by $f(x), f(y)$ and $f(z)$. This structure induces a forbidden triple $(\mathrm{Id}, \mathrm{Id}, a)$ which contradicts the assumption $\mathbf{A} \in \mathrm{RRA}$. □

Recall from Section 2.6.3 the terminology of edges with certain types that is due to Bulatov (see, e.g., [Bul16]). We introduce the following notation for a clone of canonical polymorphisms.

4.38 Definition We call a subset $\{a, b\} \subseteq A_0$ an *edge of* $\mathrm{Pol}^{\mathrm{can}}(\mathfrak{B})$ and we call the elements in

$$Q := \big\{\{a, b\} \subseteq A_0 \mid \exists g \in \mathrm{Pol}^{\mathrm{can}}(\mathfrak{B}) \text{ such that } \overline{g} \text{ is symmetric on } \{a, b\}\big\}$$

the *red edges* of $\mathrm{Pol}^{\mathrm{can}}(\mathfrak{B})$.

Note that the red edges of $\mathrm{Pol}^{\mathrm{can}}(\mathfrak{B})$ are by Proposition 4.14 precisely the semilattice edges of $\mathrm{Pol}(\mathfrak{A}_0)$, where $\mathfrak{A}_0$ is the atom structure of $\mathbf{A}$. With this observation in mind, the next lemma follows immediately from Proposition 2.42. We include the proof here because it illustrates a useful idea of composing conservative operations.

4.39 Lemma *There exists a binary canonical polymorphism that is symmetric on all red edges and behaves on each non-red edge like a projection. We call this function* maximal-symmetric.

Proof: For each $\{a, b\} \in Q$ let $f_{a,b}$ be a canonical polymorphism such that its behaviour is symmetric on $\{a, b\}$. We prove the lemma by an induction on the size of subsets of Q, i.e., we show that for every subset $F \subseteq Q$ of size n there exists a polymorphism $f_F \in \mathrm{Pol}^{\mathrm{can}}(\mathfrak{B})$ that is symmetric on all edges from F. For each subset $\{a\}$ of Q of size one, there exists by the definition of red edges a canonical polymorphism $f_{a,a}$ with a behaviour that is symmetric on $\{a\}$. Let $F \subseteq Q$ and suppose there exists a canonical polymorphism g with symmetric behaviour on elements from F. Let $\{a_1, a_2\} \in Q \backslash F$. We want to show that there exists a canonical polymorphism with a behaviour that is symmetric on all elements from $F \cup \{a_1, a_2\}$. We may assume that this does not hold for g, otherwise we are done. Therefore, and since g is edge-conservative, we have

$$\overline{g}(a_1, a_2) \neq \overline{g}(a_2, a_1) \text{ and } \overline{g}(a_1, a_2),\ \overline{g}(a_2, a_1) \in \{a_1, a_2\}.$$

With this it is easy to see that $f_{a_1,a_2}(g(x, y), g(y, x))$ is a polymorphism with a behaviour that is symmetric on all elements from $F \cup \{a_1, a_2\}$.

This proves the first part of the statement. For the second part note that for a binary canonical edge-conservative polymorphism there are only 4 possibilities for the behaviour on a set $\{a_1, a_2\}$. If $\{a_1, a_2\}$ is not a red edge then every binary canonical edge-conservative polymorphism behaves like a projection on $\{a_1, a_2\}$. □

4.6.3 Canonical Ternary Polymorphisms

We obtain in this section a result that states that the existence of a ternary $\{a,b\}$-canonical polymorphism f implies the existence of a canonical polymorphism with the same behaviour on $\{a,b\}$ as f (Corollary 4.42). This is done similarly as in Section 4.6.2.

4.40 Lemma *Let $s' \in \mathrm{Pol}^{\mathrm{can}}(\mathfrak{B})$ be an injective, maximal-symmetric polymorphism. Then the function $s^*\colon \mathfrak{B}^3 \to \mathfrak{B}^3$ where $s^*(x_1,x_2,x_3)$ is defined by*

$$(s'(s'(x_1,x_2),s'(x_2,x_3)),s'(s'(x_2,x_3),s'(x_3,x_1)),s'(s'(x_3,x_1),s'(x_1,x_2)))$$

is a homomorphism. Moreover, for all $x,y \in B^3$ with $x \neq y$ it holds that

$$\overline{\mathrm{Id}}^{\mathfrak{B}^3}(s^*(x),s^*(y)).$$

Note that this means that two distinct tuples in the image of s^* have distinct entries in each coordinate.

Proof: Let $x,y \in B^3$ and suppose that $a^{\mathfrak{B}^3}(x,y)$ holds for $a \in A$. By the definition of the product structure $a^{\mathfrak{B}}(x_i,y_i)$ holds for all $i \in \{1,2,3\}$. Since s' is a polymorphism clearly $a^{\mathfrak{B}}(s^*(x)_i,s^*(y)_i)$ holds by the definition of s^*. Now we use again the definition of a product structure and get $a^{\mathfrak{B}^3}(s^*(x),s^*(y))$ which shows that s^* is a homomorphism.

For the second part of the statement let $x,y \in B^3$ distinct. Suppose that $a_1^{\mathfrak{B}}(x_1,y_1)$, $a_2^{\mathfrak{B}}(x_2,y_2)$ and $a_3^{\mathfrak{B}}(x_3,y_3)$ hold for some $a_1,a_2,a_3 \in A_0$, where at least one atom is different from Id. Since s' is injective we have

$$\overline{\mathrm{Id}}^{\mathfrak{B}}(s'(x_1,x_2),s'(y_1,y_2)) \text{ or } \overline{\mathrm{Id}}^{\mathfrak{B}}(s'(x_2,x_3),s'(y_2,y_3)).$$

Again by the injectivity of s' we get

$$\overline{\mathrm{Id}}^{\mathfrak{B}}(s'(s'(x_1,x_2),s'(x_2,x_3)),s'(s'(y_1,y_2),s'(y_2,y_3))).$$

By the definition of s^* this shows that $\overline{\mathrm{Id}}^{\mathfrak{B}}(s^*(x)_1,s^*(y)_1)$ holds. It is easy to see by analogous arguments that the same is true for the other coordinates. Therefore, the statement follows. □

4.41 Proposition *Let $a \nleq \mathrm{Id}$ and $b \nleq \mathrm{Id}$ be atoms of $\mathbf{A}$ such that $\{a,b\} \notin Q$. Let $m \in \mathrm{Pol}(\mathfrak{B})$ be ternary $\{a,b\}$-canonical and $s' \in \mathrm{Pol}(\mathfrak{B})$ be injective, maximal-symmetric. Then there exists a canonical $m' \in \mathrm{Pol}(\mathfrak{B})$ with the same behaviour as m on $\{a,b\}$.*

Proof: By Lemma 4.39 we may assume that s' behaves on $\{a,b\}$ like the projection to the first coordinate since $\{a,b\} \notin Q$. Let s^* be the function defined in Lemma 4.40 and consider the function $m'\colon B^3 \to B$ which is defined by $m^*(x) := m(s^*(x))$.

<u>Claim 1:</u> m^* is injective.

Let $x, y \in B^3$ be two distinct elements. By Lemma 4.40 we know that

$$\overline{\mathrm{Id}}^{\mathfrak{B}^3}(s^*(x), s^*(y))$$

holds. Since m is a polymorphism of $\mathfrak{B}$ we directly get that

$$\overline{\mathrm{Id}}^{\mathfrak{B}}(m^*(x), m^*(y))$$

holds, which proves the injectivity of m^*.

<u>Claim 2:</u> m^* is $\{a, b\}$-canonical and behaves on $\{a, b\}$ like m.

Let $x, y \in B^3$ with $(q_1, q_2, q_3)(x, y)$ such $q_1, q_2, q_3 \in \{a, b\}$. Since s behaves like the first projection on $\{a, b\}$ it follows that $(q_1, q_2, q_3)(s^*(x), s^*(y))$. Together with the $\{a, b\}$-canonicity of m this proves Claim 2.

Since m^* is injective there exists by Proposition 4.32 a polymorphism $m^*_<$ of $\mathfrak{B}_<$. Since $\mathfrak{B}_<$ is a Ramsey structure we can apply Theorem 2.53 to $m^*_<$. Let g be the resulting polymorphism that is canonical with respect to $\mathfrak{B}_<$. Note that if we consider g as a polymorphism of $\mathfrak{B}$ it behaves on $\{a, b\}$ like m^* and therefore like m. Now we consider the induced behaviour of g on all 2-orbits that satisfy $x < y$. Since all atoms of $\mathbf{A}$ are symmetric and $\overline{g}$ is conservative this induces a function $h\colon (A_0 \backslash \{\mathrm{Id}\})^3 \to A_0 \backslash \{\mathrm{Id}\}$.

<u>Claim 3:</u> The partial behaviour h does not induce a forbidden triple.

Assume for contradiction that there exist $x, y, z \in B^3$ such that the application of an operation with behaviour h would induce a forbidden triple.Without loss of generality we can order the elements of each coordinate of x, y, z strictly with $x_i < y_i < z_i$ for $i \in \{1, 2, 3\}$. Note that if on some coordinate there would be the relation Id then we are out of the domain of the behaviour h.

If we choose such an order we can find isomorphic copies $\mathfrak{A}$ of this structure (with the order) in $\mathfrak{B}_<$. If we apply the polymorphism g to this copy and forget the order of the structure $g(\mathfrak{A})$ we get a structure that is by definition isomorphic to the forbidden triple.This proves Claim 3.

To finish the proof of the lemma note that the composition of s^* with the projection to the i-th coordinate for $i \in \{1, 2, 3\}$ is a canonical, injective polymorphism (for injectivity see Lemma 4.40) and therefore induces a behaviour $f_i\colon A_0^3 \to A_0 \backslash \{\mathrm{Id}\}$. We define $f\colon A_0^3 \to (A_0 \backslash \{\mathrm{Id}\})^3$ by $f(a_1, a_2, a_3) := (f_1(a_1), f_2(a_2), f_3(a_3))$. The composition $h \circ f\colon A_0^3 \to A_0$ is the behaviour of a canonical function of $\mathfrak{B}$. If $h \circ f$ would induce a forbidden triple then also h would induce a forbidden triple, which contradicts Claim 3. □

4.42 Corollary *Let $\mathfrak{B}$ have a binary injective polymorphism. Let $a, b \in A_0$ be such that no $\{a, b\}$-symmetric polymorphism exists. Let m be a ternary $\{a, b\}$-canonical polymorphism. Then there exists a canonical polymorphism m' with the same behaviour on $\{a, b\}$ as m.*

Proof: By assumption there is no canonical $\{a, b\}$-symmetric polymorphism and therefore $\{a, b\} \notin Q$. By Corollary 4.34 $\mathfrak{B}$ has a canonical binary injective polymorphism. This polymorphism is a witness that $\{c, \mathrm{Id}\} \in Q$ for all $c \in A_0 \backslash \{\mathrm{Id}\}$. With Lemma 4.39 we get a maximal symmetric polymorphism h. Since $\{c, \mathrm{Id}\} \in Q$ we get that h is $\{c, \mathrm{Id}\}$-symmetric for all $c \in A_0$. By Observation 4.37 it follows $\overline{h}(c, \mathrm{Id}) = c = \overline{h}(\mathrm{Id}, c)$, which implies that h is injective. Now Proposition 4.41 implies the statement. □

4.7 The Independence Lemma and How To Use It

The central result of this section is Proposition 4.44 which states that the absence of an $\{a, b\}$-symmetric polymorphism implies that all polymorphism are canonical on $\{a, b\}$. The main ingredients of our proof of this proposition are the fact that $\mathbf{A} \in \mathrm{RRA}$ has a flexible atom and the following "Independence Lemma" (Lemma 4.43).

4.7.1 The Independence Lemma

The following lemma transfers the absence of a special partially canonical polymorphism to the existence of certain relations of arity 4 that are primitively positively definable in $\mathfrak{B}$. A lemma of a similar type appeared as Lemma 42 in an article by [BP14].

4.43 Lemma (Independence Lemma) *Let $\mathfrak{B}$ be a homogeneous structure with finite relational signature. Let a and b be 2-orbits of $\mathrm{Aut}(\mathfrak{B})$ such that a, b, and $(a \cup b)$ are primitively positively definable in $\mathfrak{B}$. Then the following are equivalent:*

1. *$\mathfrak{B}$ has an $\{a, b\}$-canonical polymorphism g that is $\{a, b\}$-symmetric with $\overline{g}(a, b) = \overline{g}(b, a) = a$.*

2. *For every primitive positive formula φ such that $\varphi \wedge a(x_1, x_2) \wedge b(y_1, y_2)$ and $\varphi \wedge b(x_1, x_2) \wedge a(y_1, y_2)$ are satisfiable over $\mathfrak{B}$, the formula $\varphi \wedge a(x_1, x_2) \wedge a(y_1, y_2)$ is also satisfiable over $\mathfrak{B}$.*

3. *For every finite $F \subset B^2$ there exists a homomorphism h_F from the substructure of $\mathfrak{B}^2$ induced by F to $\mathfrak{B}$ that is $\{a, b\}$-canonical with $\overline{h_F}(a, b) = \overline{h_F}(b, a) = a$.*

Proof: The implication from (1) to (2) follows directly by applying the symmetric polymorphisms to tuples from the relation defined by φ.

For the implication from (2) to (3) let F be a finite subset of B^2. Let $\{e_1, \ldots, e_n\}$ with $n \in \mathbb{N}$ be an enumeration of F. To construct h_F consider the formula φ_0 with variables $x_{i,j}$ for $1 \leqslant i, j \leqslant n$ that is the conjunction of all atomic formulas $R(x_{i_1,j_1}, \ldots, x_{i_k,j_k})$ such that $R(e_{i_1}, \ldots, e_{i_k})$ and $R(e_{j_1}, \ldots, e_{j_k})$ hold in $\mathfrak{B}$. Note that this formula states exactly which relations hold on F in $\mathfrak{B}^2$. Let P be the set of pairs $((i_1, i_2), (j_1, j_2))$ such that

$$
\begin{aligned}
&(a \cup b)(e_{i_1}, e_{i_2}) \\
\text{and } &(a \cup b)(e_{j_1}, e_{j_2}) \\
\text{and } &(a(e_{i_1}, e_{i_2}) \vee a(e_{j_1}, e_{j_2})) \\
\text{and } &(b(e_{i_1}, e_{i_2}) \vee b(e_{j_1}, e_{j_2})).
\end{aligned}
$$

If we show that the formula

$$\psi := \varphi_0 \wedge \bigwedge_{((i_1,i_2),(j_1,j_2)) \in P} a(x_{i_1,j_1}, x_{i_2,j_2})$$

is satisfiable by an assignment α, we get the desired homomorphism by setting $h_F(e_i, e_j) := \alpha(x_{i,j})$. We prove the satisfiability of ψ by induction over the size of subsets I of P. For the inductive beginning consider an element $((i_1, i_2), (j_1, j_2)) \in P$. Without loss of generality we have that $a(i_1, i_2)$ holds. Therefore the assignment $\alpha(x_{i,j}) := e_i$ witnesses the satisfiability of the formula $\varphi_0 \wedge a(x_{i_1,j_1}, x_{i_2,j_2})$. For the inductive step let $I \subseteq P$ be of size m and assume that the statement is true for subsets of size $m-1$. Let $p_1 = ((u_1, u_2), (v_1, v_2))$ and $p_2 = ((u'_1, u'_2), (v'_1, v'_2))$ be two elements from I. We define the following formula

$$\psi_0 := \varphi_0 \wedge \bigwedge_{((i_1,i_2),(j_1,j_2)) \in I \setminus \{p_1,p_2\}} a(x_{i_1,j_1}, x_{i_2,j_2}).$$

Then by the inductive assumption the formulas $\psi_0 \wedge a(x_{u_1,v_1}, x_{u_2,v_2})$ and $\psi_0 \wedge a(x_{u'_1,v'_1}, x_{u'_2,v'_2})$ are satisfiable. The assumptions on the elements in P give us that also

$$\psi_0 \wedge a(x_{u_1,v_1}, x_{u_2,v_2}) \wedge b(x_{u'_1,v'_1}, x_{u'_2,v'_2})$$

and

$$\psi_0 \wedge b(x_{u_1,v_1}, x_{u_2,v_2}) \wedge a(x_{u'_1,v'_1}, x_{u'_2,v'_2})$$

are satisfiable; since $a \cup b$ is a primitive positive definable relation we are done otherwise. But then we can apply the assumption of (2) and get that also

$$\psi_0 \wedge a(x_{u_1,v_1}, x_{u_2,v_2}) \wedge a(x_{u'_1,v'_1}, x_{u'_2,v'_2})$$

is satisfiable, which proves the inductive step.

The direction from (3) to (1) is a standard application of König's tree lemma. For a reference see for example Lemma 42 in the article by [BP14]. □

4.7.2 Absence of $\{a, b\}$-symmetric Polymorphisms

We are now able to prove the main result of this section, which will be a corner stone in the proof of Theorem 4.1. Our proof of this proposition makes use of a 4-ary relation $E_{a,b}$ with the following first-order definition:

$$(x_1, x_2, x_3, x_4) \in E_{a,b} :\Leftrightarrow ((a \cup b)(x_1, x_2) \wedge (a \cup b)(x_3, x_4) \wedge a(x_1, x_2) \Leftrightarrow a(x_3, x_4)).$$

It is an easy observation that the $\{a, b\}$-canonical polymorphisms of $\mathfrak{B}$ are precisely those that preserve the relation $E_{a,b}$. By Theorem 2.45 we get that whenever $E_{a,b}$ is primitively positively definable in $\mathfrak{B}$ then all polymorphisms of $\mathfrak{B}$ preserve $E_{a,b}$ and are therefore $\{a, b\}$-canonical. In the following proof we use the 4-ary relations that are provided by the second item of the Independence Lemma 4.43 to provide a primitive positive definition of $E_{a,b}$.

4.44 Proposition *Let $\mathfrak{B}$ be a normal representation of a finite integral symmetric relation algebra with a flexible atom s. Suppose that $\mathfrak{B}$ has a binary injective polymorphism. Let $a \nleq \mathrm{Id}$ and $b \nleq \mathrm{Id}$ be two atoms such that $\mathfrak{B}$ has no $\{a,b\}$-symmetric polymorphism. Then all polymorphisms are canonical on $\{a,b\}$.*

Proof: By Corollary 4.34 there exists a canonical binary injective polymorphism of $\mathfrak{B}$. Therefore, for every $a' \in A_0$ the edge $\{a', \mathrm{Id}\}$ is red and the maximal symmetric polymorphism t that exists by Lemma 4.39 is symmetric on all these edges. Observation 4.37 implies that t is injective. Note that t behaves like a projection on $\{a,b\}$ since there exists no $\{a,b\}$-symmetric polymorphism.

Let ψ be the formula defined as follows:

$$\psi(x_1,x_2,y_1,y_2) := \overline{\mathrm{Id}}(x_1,y_1) \wedge \overline{\mathrm{Id}}(x_1,y_2) \wedge \overline{\mathrm{Id}}(x_2,y_1) \wedge \overline{\mathrm{Id}}(x_2,y_2).$$

We use ψ to formulate and prove the following claim:

<u>Claim 1:</u> a) There exists a formula $\varphi_a(x_1,x_2,y_1,y_2)$ such that

$$\begin{aligned}
&\varphi_a \wedge \psi(x_1,x_2,y_1,y_2) \wedge a(x_1,x_2) \wedge b(y_1,y_2) \text{ is satisfiable in } \mathfrak{B},\\
&\varphi_a \wedge \psi(x_1,x_2,y_1,y_2) \wedge b(x_1,x_2) \wedge a(y_1,y_2) \text{ is satisfiable in } \mathfrak{B},\\
&\varphi_a \wedge a(x_1,x_2) \wedge a(y_1,y_2) \text{ is not satisfiable in } \mathfrak{B}.
\end{aligned}$$

b) There exists a formula $\varphi_b(x_1,x_2,y_1,y_2)$ that has the same property with a and b in exchanged roles.

Proof. There exists a formula φ''_a that witnesses the negation of (2) in the Independence Lemma (Lemma 4.43) since $\mathfrak{B}$ does not have an $\{a,b\}$-symmetric polymorphism h with $\overline{h}(a,b) = a$. Let φ''_b be the formula that witnesses in the same way the non-existence of an $\{a,b\}$-symmetric polymorphism h of $\mathfrak{B}$ with $\overline{h}(a,b) = b$. We define for $c \in \{a,b\}$ the formula φ'_c as follows:

$$\varphi'_c(x_1,x_2,y_1,y_2) := \varphi''_c(x_1,x_2,y_1,y_2) \wedge (a \cup b)(x_1,x_2) \wedge (a \cup b)(y_1,y_2). \tag{4.2}$$

If φ'_a and φ'_b witness a) and b) in Claim 1, we are done. So suppose that they do not. Note that if we have $c \in \{a,b\}$ and $d \in \{a,b\} \backslash \{c\}$ such that

$$\varphi'_c(x_1,x_2,y_1,y_2) \wedge c(x_1,x_2) \wedge d(y_1,y_2) \wedge \mathrm{Id}(x_2,y_1) \text{ is satisfiable in } \mathfrak{B} \text{ and} \tag{4.3}$$

$$\varphi'_c(x_1,x_2,y_1,y_2) \wedge d(x_1,x_2) \wedge c(y_1,y_2) \wedge \overline{\mathrm{Id}}(x_2,y_1) \text{ is satisfiable in } \mathfrak{B}, \tag{4.4}$$

then φ'_c would satisfy the statement about φ_c in Claim 1, a) or in Claim 1, b). To see this note that we can apply the injective, maximal symmetric polymorphism t that behaves like a

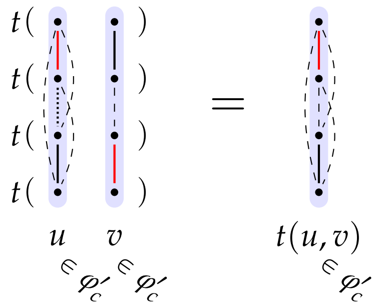


Figure 4.2: Application of the injective operation t on tuples u and v. Red and black edges correspond to atoms c and d, the dotted edge to atom Id, and the dashed lines denote $\overline{\mathrm{Id}}$.

projection on $\{a, b\}$ to the tuples u and v that witness (4.3) and (4.4). The first tuple satisfies $\overline{\mathrm{Id}}(x_1, y_1)$, $\overline{\mathrm{Id}}(x_2, y_2)$ and $\overline{\mathrm{Id}}(x_1, y_2)$ since $\mathrm{Id}(x_2, y_1)$ and $c \neq d$. The second tuple satisfies $\overline{\mathrm{Id}}(x_2, y_1)$. Then the injectivity of t ensures that the tuples $t(u, v)$ and $t(v, u)$ witness Claim 1, a) or Claim 1, b). Figure 4.2 illustrates this situation.

By our assumption that Claim 1 is not satisfied by φ'_a and φ'_b we conclude that for at least one $c \in \{a, b\}$ it holds that for $d \in \{a, b\} \setminus \{c\}$

$$\varphi'_c(x_1, x_2, y_1, y_2) \wedge c(x_1, x_2) \wedge d(y_1, y_2) \wedge \mathrm{Id}(x_2, y_1) \text{ is satisfiable in } \mathfrak{B} \text{ and} \tag{4.5}$$

$$\varphi'_c(x_1, x_2, y_1, y_2) \wedge d(x_1, x_2) \wedge c(y_1, y_2) \wedge \mathrm{Id}(x_2, y_1) \text{ is satisfiable in } \mathfrak{B}. \tag{4.6}$$

We distinguish the following different cases.

1. φ'_a satisfies a) in Claim 1 and (4.5) and (4.6) hold for $c = b$ and $d = a$.

2. (4.5) and (4.6) hold for $c = a$ and $d = b$ and φ'_b satisfies b) in Claim 1.

3. (4.5) and (4.6) hold for $c = a$ and $d = b$ as well as for $c = b$ and $d = a$.

Case 1: Consider the following formula φ_b with

$$\varphi_b(x_1, x_2, y_1, y_2) := \exists z_1, z_2.(\varphi'_b(x_1, x_2, x_2, z_1) \wedge \varphi'_a(x_2, z_1, z_2, y_1) \wedge \varphi'_b(z_2, y_1, y_1, y_2)).$$

We claim that φ_b satisfies b) in Claim 1. This proves Claim 1, because φ'_a satisfies a) in Claim 1. To see that φ_b fulfills the two satisfiability statements in Claim 1, b) we can first amalgamate the structure $\mathfrak{B}_1$ induced on the elements of a satisfying tuple for φ'_b with the structure $\mathfrak{B}_2$ induced by the elements of a satisfying tuple for φ'_a. We amalgamate these two structures over their common substructure $\mathfrak{A}$ induced by the variables x_2 and z_1, with the variable names from the definition of φ_b. In this amalgamation step all missing edges are

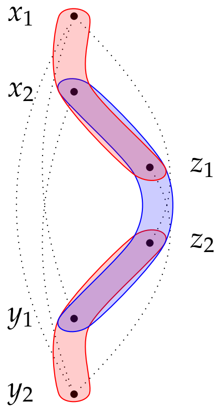


Figure 4.3: φ_b build from φ'_b (red), φ'_a (blue), and the flexible atom s (dotted).

filled with the flexible atom s. In a second amalgamation step we amalgamate the resulting structure with another copy of the structure $\mathfrak{B}_1$, but now with the common substructure on the variables z_1 and y_1 (again with refer to the names used in the definition of φ_b). As before the missing edges are filled with the flexible atom s. Figure 4.3 illustrates the situation. It follows from the choice of φ'_a and φ'_b and the definition of φ_a that $\varphi_a \wedge a(x_1, x_2) \wedge a(y_1, y_2)$ is not satisfiable in $\mathfrak{B}$.

Case 2: This case is analogous to Case 1.

Case 3: Consider the formula φ_a with

$$\varphi_a(x_1, x_2, y_1, y_2) := \exists z. (\varphi'_a(x_1, x_2, x_2, z) \wedge \varphi'_b(x_2, z, z, y_1) \wedge \varphi'_a(z, y_1, y_1, y_2)).$$

We show that $\varphi_a \wedge \psi(x_1, x_2, y_1, y_2) \wedge a(x_1, x_2) \wedge b(y_1, y_2)$ is satisfiable in $\mathfrak{B}$. Since (4.5) holds for $c = a$ and $d = b$ and since a and b are distinct, there exists $p_1 \in A_0 \backslash \{\mathrm{Id}\}$ such that

$$\varphi'_a(x_1, x_2, y_1, y_2) \wedge a(x_1, x_2) \wedge b(y_1, y_2) \wedge \mathrm{Id}(x_2, y_1) \wedge p_1(x_1, y_2)$$

is satisfiable in $\mathfrak{B}$. Similarly, since (4.5) holds for $c = b$ and $d = a$, there exists $p_2 \in A_0 \backslash \{\mathrm{Id}\}$ such that

$$\text{and } \varphi'_b(x_1, x_2, y_1, y_2) \wedge b(x_1, x_2) \wedge a(y_1, y_2) \wedge \mathrm{Id}(x_2, y_1) \wedge p_2(x_1, y_2).$$

Note that there are $u_1, \ldots, u_5 \in B$ such that the following atomic formulas hold:

$$\begin{aligned} & a(u_1, u_2), p_1(u_1, u_3), s(u_1, u_4), s(u_1, u_5), \\ & \qquad b(u_2, u_3), p_2(u_2, u_4), s(u_2, u_5), \end{aligned}$$

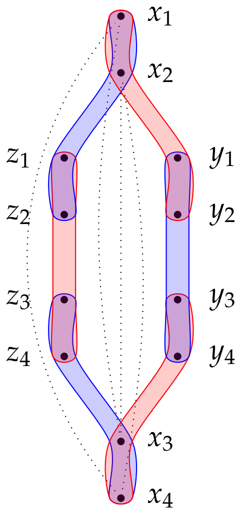


Figure 4.4: The formula δ build from φ'_a (red), φ'_b (blue), and the flexible atom s (dotted).

$$a(u_3, u_4), p_1(u_3, u_5),$$
$$b(u_4, u_5).$$

If we choose for the existentially quantified variable z in the definition of φ_a the element u_3 then the tuple (u_1, u_2, u_4, u_5) satisfies the formula

$$\varphi_a \wedge \psi(x_1, x_2, y_1, y_2) \wedge a(x_1, x_2) \wedge b(y_1, y_2).$$

By an analogous argument also $\varphi_a \wedge \psi(x_1, x_2, y_1, y_2) \wedge b(x_1, x_2) \wedge a(y_1, y_2)$ is satisfiable. It follows again from the choice of φ'_a and φ'_b and the definition of φ_a that $\varphi_a \wedge a(x_1, x_2) \wedge a(y_1, y_2)$ is not satisfiable in $\mathfrak{B}$. By an analogous definition we can find a formula φ_b that satisfies b) in Claim 1. Therefore, we are done with Case 3. Altogether this proves Claim 1.

Let φ_a and φ_b be the two formulas that exist by Claim 1. We define the following formulas

$$\varphi'_a(x_1, x_2, y_1, y_2) := \varphi_a(x_1, x_2, y_1, y_2) \wedge (a \cup b)(x_1, x_2) \wedge (a \cup b)(y_1, y_2)$$
$$\varphi'_b(x_1, x_2, y_1, y_2) := \varphi_b(x_1, x_2, y_1, y_2) \wedge (a \cup a)(x_1, x_2) \wedge (a \cup b)(y_1, y_2).$$

We also define a formula δ as follows (see also Figure 4.4):

$$\begin{aligned}\delta(x_1, x_2, x_3, x_4) := {} & s(x_1, x_3) \wedge s(x_1, x_4) \wedge s(x_2, x_3) \wedge s(x_2, x_4) \\ & \wedge \exists y_1, y_2, y_3, y_4.(\varphi'_a(x_1, x_2, y_1, y_2) \wedge \varphi'_b(y_1, y_2, y_3, y_4) \\ & \wedge \varphi'_a(y_3, y_4, x_3, x_4))\end{aligned}$$

$$\wedge\ \exists z_1, z_2, z_3, z_4.(\varphi'_b(x_1, x_2, z_1, z_2) \wedge \varphi'_a(z_1, z_2, z_3, z_4)$$
$$\wedge\ \varphi'_b(z_3, z_4, x_3, x_4)).$$

Analogously to Case 1, an amalgam of the structures that are induced by tuples that satisfy φ'_a and φ'_b shows that the formulas $\delta \wedge a(x_1, x_2) \wedge b(x_3, x_4)$ and $\delta \wedge b(x_1, x_2) \wedge a(x_3, x_4)$ are satisfiable in $\mathfrak{B}$. Note that this is possible since we ensured in Claim 1 that there exist tuples that additionally satisfy ψ. It also holds that a tuple x that satisfies δ also satisfies

$$(a(x_1, x_2) \wedge b(x_3, x_4)) \vee (b(x_1, x_2) \wedge a(x_3, x_4)). \tag{4.7}$$

Assume that for a tuple x that satisfies δ it holds that $a(x_1, x_2) \wedge a(x_3, x_4)$. Then there exist y_1, y_2, y_3, y_4 such that $\varphi'_a(x_1, x_2, y_1, y_2) \wedge \varphi'_a(y_3, y_4, x_3, x_4)$ holds. But this is by the definition of φ'_a only possible if $b(y_1, y_2)$ and $b(y_3, y_4)$ hold, in contradiction to $\varphi'_b(y_1, y_2, y_3, y_4)$. The same argument works for proving that $\neg(b(x_1, x_2) \wedge b(x_3, x_4))$ holds.

We complete the proof with a primitive positive definition of $E_{a,b}$. We have the following primitive positive formula

$$\delta'(x_1, x_2, x_3, x_4) := \exists y_1, y_2.(\delta(x_1, x_2, y_1, y_2) \wedge \delta(y_1, y_2, x_3, x_4)),$$

and define $E_{a,b}$ by

$$(x_1, x_2, x_3, x_4) \in E_{a,b} \Leftrightarrow \delta'(x_1, x_2, x_3, x_4).$$

For the forward direction of this equivalence, let x be a tuple from $E_{a,b}$ such that $c(x_1, x_2)$ holds for $c \in \{a, b\}$. Let $d \in \{a, b\} \backslash \{c\}$ and let y_1 and y_2 be two elements from $\mathfrak{B}$ such that $d(y_1, y_2)$ and $s(x_i, y_j)$ for every $i \in \{1, \ldots, 4\}$ and every $j \in \{1, 2\}$ holds. Such elements exists since in the substructure of $\mathfrak{B}$ that is induced by $\{x_1, x_2, x_3, x_4, y_1, y_2\}$ all appearing triangles are allowed by the definition of the flexible atom s. The elements y_1 and y_2 witness that x satisfies the formula δ'.

For the other direction assume that a tuple x satisfies δ'. Then there exist y_1 and y_2 such that $\delta(x_1, x_2, y_1, y_2) \wedge \delta(y_1, y_2, x_3, x_4)$ is satisfied. Since we observed that the tuples (x_1, x_2, y_1, y_2) and (y_1, y_2, x_3, x_4) both satisfy (4.7) we can assume that $c(x_1, x_2)$ holds for $c \in \{a, b\}$. It follows also from (4.7) that $d(y_1, y_2)$ holds for $d \in \{a, b\} \backslash \{c\}$ and then (4.7) implies that $c(x_3, x_4)$ holds, which proves the backward direction of the stated equivalence. □

4.8 Proof of the Result

In this section we prove the main results of this article. We first obtain a dichotomy theorem for a class of CSPs (Theorem 4.45). This is used in combination with the observations in Section 4.2 to conclude the proof of Theorem 4.1.

4.45 Theorem *Let* $\mathbf{A} \in \mathrm{RRA}$ *be finite integral symmetric and with a flexible atom s and let* A_0 *be the set of atoms of* $\mathbf{A}$*. Then either*

- *there exists an operation* $f\colon A_0^6 \to A_0$ *that preserves the allowed triples of* $\mathbf{A}$*, satisfies*

$$\forall x_1, \dots, x_6 \in A_0.\, f(x_1, \dots, x_6) \in \{x_1, \dots x_6\}$$

  and satisfies the Siggers identity

$$\forall x, y, z \in A_0.\, f(x, x, y, y, z, z) = f(y, z, x, z, x, y);$$

  in this case, $\mathrm{CSP}(\mathfrak{B})$ *is in P, or*

- $\mathrm{HSP}^{\mathrm{fin}}(\{\mathrm{Pol}(\mathfrak{B})\})$ *contains a 2-element algebra where all operations are projections; in this case,* $\mathrm{CSP}(\mathfrak{B})$ *is NP-complete.*

Proof: Let $\mathfrak{B}$ be a normal representation of $\mathbf{A}$ that exists by Proposition 4.6. The finite boundedness of $\mathfrak{B}$ implies that $\mathrm{CSP}(\mathfrak{B})$ is in NP. Let $\mathfrak{A}_0$ be the atom structure of $\mathbf{A}$. We can assume that $\mathfrak{B}$ has a binary injective polymorphism, because otherwise Proposition 4.24 would directly imply the second item. The existence of a binary injective polymorphism implies by Corollary 4.34 the existence of a canonical binary injective polymorphism g.

If the first item of the theorem is satisfied then the operation f is a Siggers polymorphism of $\mathfrak{A}_0$ and the statement follows by the Propositions 4.14 and 4.15.

Assume therefore that the first item in the theorem does not hold. By Corollary 4.18 there exist elements $a, b \in A_0$ such that the subalgebra of $\mathrm{Pol}(\mathfrak{A}_0)$ on $\{a, b\}$ contains only projections. It holds that $\mathrm{Id} \notin \{a, b\}$, since g is a witness that Id can not be in the domain of a subalgebra that contains only projections. Since all operations from $\mathrm{Pol}(\mathfrak{A}_0)$ are projections on $\{a, b\}$ there exists no canonical polymorphism of $\mathfrak{B}$ that is $\{a, b\}$-symmetric. By Proposition 4.36 there exists also no $\{a, b\}$-symmetric polymorphism of $\mathfrak{B}$. Since $\mathfrak{B}$ has a binary injective polymorphism we can apply Proposition 4.44 and get that all polymorphisms of $\mathfrak{B}$ are $\{a, b\}$-canonical. The last step is to show that all polymorphisms of $\mathfrak{B}$ behave like projections on $\{a, b\}$.

Assume for contradiction that there exists a ternary, $\{a, b\}$-canonical polymorphism m that behaves on $\{a, b\}$ like a majority or like a minority. By Corollary 4.42 there exists a canonical polymorphism that is also a majority or minority on $\{a, b\}$ (here we use again the existence of an injective polymorphism). This contradicts our assumption that $\mathrm{Pol}(\mathfrak{A}_0)$ is trivial on $\{a, b\}$. We get that every polymorphism of $\mathfrak{B}$ does not behave on $\{a, b\}$ as an operation from Post's theorem (Theorem 2.40) and therefore must behave as a projection on $\{a, b\}$ by Theorem 2.40. Thus, $\mathrm{HSP}^{\mathrm{fin}}(\{\mathrm{Pol}(\mathfrak{B})\})$ contains a 2-element algebra whose operations are projections and $\mathrm{CSP}(\mathfrak{B})$ is NP-hard, according to Theorem 2.47. □

We can prove the main result.

Proof of Theorem 4.1: Let $\mathbf{A}$ be as in the assumptions of the theorem and let $\mathbf{A}'$ be the finite symmetric integral representable relation algebra that exists by Proposition 4.4. Suppose that $\mathbf{A}$ satisfies the first condition of Theorem 4.1. By Item 3) in Proposition 4.4, we get that then $\mathbf{A}'$ satisfies the first condition in Theorem 4.45 and therefore $\mathrm{CSP}(\mathfrak{B})$ for the normal representation $\mathfrak{B}$ of $\mathbf{A}'$ is in P. By Section 2.3.5 we know that $\mathrm{NSP}(\mathbf{A}')$ and $\mathrm{CSP}(\mathfrak{B})$ are polynomial-time equivalent. This, together with the Turing reduction from $\mathrm{NSP}(\mathbf{A})$ to $\mathrm{NSP}(\mathbf{A}')$ by Item 1) in Proposition 4.4 implies that $\mathrm{NSP}(\mathbf{A})$ is in P. This proves the first part of the theorem.

Assume that $\mathbf{A}$ does not satisfy the first condition of Theorem 4.1. Item 3) in Proposition 4.4 again implies that $\mathbf{A}'$ does not satisfy the first condition in Theorem 4.45 and therefore $\mathrm{CSP}(\mathfrak{B})$ for the normal representation $\mathfrak{B}$ of $\mathbf{A}'$ is NP-complete. As before we get that $\mathrm{NSP}(\mathbf{A}')$ is NP-complete and the many-one reduction from $\mathrm{NSP}(\mathbf{A})$ to $\mathrm{NSP}(\mathbf{A}')$ by Item 2) in Proposition 4.4 implies the NP-hardness of $\mathrm{NSP}(\mathbf{A})$. The containment in NP follows by Item 1) in Proposition 4.4. This concludes the proof of Theorem 4.1. □

4.46 Corollary *For a given finite, symmetric* $\mathbf{A} \in \mathrm{RRA}$ *with a flexible atom it is decidable which of the two items in Theorem 4.1 holds. In particular, it is decidable whether* $\mathrm{NSP}(\mathbf{A})$ *is solvable in polynomial time.*

Proof: Since A_0 is a finite set one can go through all possible operations $f\colon A_0^6 \to A_0$ that preserve all the allowed triples of $\mathbf{A}$ and check whether the Siggers identity is satisfied. If $\mathrm{P} \neq \mathrm{NP}$, it follows from Theorem 4.1 that it is decidable whether $\mathrm{NSP}(\mathbf{A})$ is in P. In the case of $\mathrm{P} = \mathrm{NP}$ this is a trivial task. □

The problem of deciding whether certain identities hold in the polymorphism clone of a structure is well known problem in the study of CSPs. The computational complexity is known to be in NP [CL17]. The precise complexity of deciding whether a polymorphism clone of an explicitly given structure has a conservative operation that satisfies the Siggers identity is open [CL17].

4.9 Connection to Smooth Approximations

We discuss the relationship of our results to the techniques developed by Mottet and Pinsker [MP22]. The main invention of [MP22] are *smooth approximations* which are equivalence relations on sets of n-tuples. The purpose of these equivalence relations is to *approximate* the prominent *orbit-equivalence relation*. We have seen the importance of these relations in the present article: the polymorphisms which preserve the orbit-equivalence relation are precisely the canonical ones and they store the information about possible finite-domain algorithms that can be used to solve the infinite-domain CSP (cf. Section 4.3).

We start by rearranging the results of Section 4.7 such that we get the following theorem. In order to repeat the key steps of our main proof with a focus on the similarities to [MP22] the assumptions in this theorem are a natural starting point.

4.47 Theorem *Let $\mathfrak{B}$ be a normal representation of a finite integral symmetric relation algebra with a flexible atom s and let $\mathfrak{A}_0$ be the atom structure of $\mathfrak{B}$. Suppose that $\mathrm{Pol}(\mathfrak{B})$ contains a binary injective polymorphism and $\mathrm{Pol}(\mathfrak{A}_0)$ does not have a Siggers operation. Then there exists two atoms $a \nleq \mathrm{Id}$ and $b \nleq \mathrm{Id}$ such that one of the following holds:*

1. *The orbit-equivalence relation $E_{a,b}$ is primitively positively definable in $\mathfrak{B}$.*
2. *For all $x, y \in \{a, b\}$ and every primitive positive formula φ such that $\varphi \wedge x(x_1, x_2) \wedge y(y_1, y_2)$ and $\varphi \wedge y(x_1, x_2) \wedge x(y_1, y_2)$ are satisfiable over $\mathfrak{B}$, the formula $\varphi \wedge x(x_1, x_2) \wedge x(y_1, y_2)$ is also satisfiable over $\mathfrak{B}$.*

This theorem is similar to the "Loop lemma of approximations", Theorem 10 in [MP22], even though it does not make proper use of the approximation idea ($E_{a,b}$ approximates itself). The first case in the theorem leads to the hardness condition that $\mathrm{HSP}^{\mathrm{fin}}(\{\mathrm{Pol}(\mathfrak{B})\})$ contains a 2-element algebra whose operations are projections and therefore $\mathrm{CSP}(\mathfrak{B})$ is NP-complete.

However, the second case is "stronger" than the second case in Theorem 10 from [MP22]. The strength in the statement relies on the special class of problems in our article. To see what we mean by this, we continue with the strategy of [MP22]. As a next step we can restrict the "Independence Lemma" (Lemma 4.43) as follows:

4.48 Lemma *Let $\mathfrak{B}$ be a homogeneous structure with finite relational signature. Let a and b be 2-orbits of $\mathrm{Aut}(\mathfrak{B})$ such that a, b, and $(a \cup b)$ are primitively positively definable in $\mathfrak{B}$.*

Assume that for every primitive positive formula φ such that $\varphi \wedge a(x_1, x_2) \wedge b(y_1, y_2)$ and $\varphi \wedge b(x_1, x_2) \wedge a(y_1, y_2)$ are satisfiable over $\mathfrak{B}$, the formula $\varphi \wedge a(x_1, x_2) \wedge a(y_1, y_2)$ is also satisfiable over $\mathfrak{B}$. Then $\mathfrak{B}$ has an $\{a, b\}$-canonical polymorphism g that is $\{a, b\}$-symmetric with $\overline{g}(a, b) = \overline{g}(b, a) = a$.

Note that the assumptions in this lemma are precisely what we get from Case 2) in Theorem 4.47. A lemma of similar style can be also found as Lemma 13 in the article [MP22].

They obtain in this lemma a *weakly commutative* operation. The $\{a,b\}$-symmetric operations from our article are a special case of weakly commutative operations. The difference between these two properties seems crucial for the next step of our proof: while $\{a,b\}$-symmetric operations can often be "lifted" to canonical $\{a,b\}$-symmetric operations this seems not clear in general for weakly commutative operations. This last step is necessary in order to obtain a contradiction to our assumption that $\mathrm{Pol}(\mathfrak{A}_0)$ does not have a Siggers operation.

In order to apply the results of [MP22] directly to obtain the dichotomy results of our article one would have to find a way to canonize the weakly commutative operations in a suitable way. It does not seem obvious how to do this and therefore we have to use our variant of a loop lemma (Theorem 4.47) which contains the key technical part of our result.

4.10 Conclusion and Discussion

We classified the computational complexity of the network satisfaction problem for finite symmetric $\mathbf{A} \in \mathrm{RRA}$ with a flexible atom and obtained a P versus NP-complete dichotomy. We gave a decidable criterion for $\mathbf{A}$ that is a sufficient condition for the membership of $\mathrm{NSP}(\mathbf{A})$ in P, which is also necessary (unless $\mathrm{P} = \mathrm{NP}$).

We discuss possible generalizations of this result. Of course, all generalizations go in the direction of a complete classification for NSPs of relation algebras. We give a list of open problems on various aspects of a possible continuation of our work. The first obvious generalizations arise from the omission of the assumptions on the relation algebra in Theorem 4.1.

4.49 Problem *Prove Theorem 4.1 without the assumption that the relation algebra* $\mathbf{A}$ *is symmetric.*

Many of the results from this chapter apply even without the assumption that $\mathbf{A}$ is symmetric. By Section 4.2, the relation algebra $\mathbf{A}$ with a flexible atom has a normal representation with a Ramsey expansion. Furthermore, the polynomial-time results from Section 4.3 still hold. Unfortunately, the handling of the canonical functions is more complicated, meaning that it is not clear how to obtain the results of Section 4.6 in this case.

Analogously, we could also drop the assumption that $\mathbf{A}$ has a flexible atom.

4.50 Problem *Prove that the NSP of a finite symmetric relation algebra* $\mathbf{A}$ *that does not satisfy the first item in Theorem 4.1 is NP-hard.*

A solution of this problem implies the classification result Theorem 4.1 without the assumption that $\mathbf{A}$ has a flexible atom, with the small difference that in the second item we get NP-hardness instead of NP-completeness. The reason for this difference is that symmetric relation algebras do not necessarily have normal representations and therefore containment of the NSP in NP is not longer ensured. In fact, we already mentioned examples of symmetric relation algebras with undecidable NSP. This raises the following question.

4.51 Problem *Which symmetric relation algebras have a normal representation?*

It is known that the existence of a normal representation for a finite relation algebra is decidable (see for example [BKS20]). Can this procedure, at least in the case of symmetric relation algebras, be translated into a “compact” criterion on the relation algebra, perhaps in the style of the first item of Theorem 4.1.

The next obstacle on the way to a solution of Problem 4.50 is the following:

4.52 Problem *Prove that every normal representation* $\mathfrak{B}$ *of a symmetric relation algebra* $\mathbf{A}$ *has a Ramsey expansion.*

We have seen the importance of the Ramsey expansion especially in Section 4.6.

The next step on the way to a general classification result would be to drop both assumptions about the relation algebra, the symmetry and the existence of a flexible atom at the same time. We have to remark that in this case the statement of Theorem 4.1 is false. An example for this is the Point Algebra; even though the NSP of this relation algebra is in P [VKv90], the first condition of Theorem 4.1 does not apply.

However, in the case where the relation algebra has a normal representation, there is a general condition that distinguishes the NSPs that are solvable in polynomial time from those that are NP-hard (unless P = NP). This condition is provided by the connection to infinite-domain CSPs and the universal algebraic approach. In the formulation in which we present it here, it is due to [BP20]. For more details and background we refer to the book [Bod21].

4.53 Problem *Prove the following:*
Let $\mathfrak{B}$ *be a normal representation of a finite relation algebra* $\mathbf{A}$ *such that there exists* $f, e_1, e_2 \in \mathrm{Pol}(\mathfrak{B})$ *and it holds that*

$$\forall x, y, z \in B.\, e_1(f(x, x, y, y, z, z)) = e_2(f(y, z, x, z, x, y)).$$

Then $\mathrm{NSP}(\mathbf{A})$ *is in P.*

We discussed in Section 4.9 some new developments in the study of infinite-domain CSPs. It seems very promising to explore the potential of these and other new approaches to classify NSPs and eventually gain a deeper understanding of the computational complexity of network satisfaction problems for finite relation algebras.

Chapter 5

A Datalog-Tractability Criterion

5.1 Introduction

Local consistency methods play a crucial role in the study of network satisfaction problems (see e.g. [Dün05, Hir97, LM94]). In particular, the so-called *path consistency method* is the predominant algorithm for solving NSPs in polynomial time, if this is possible. In terms of Datalog programs, solving a problem by the path consistency method means that the problem can be solved by a Datalog program of width $(2,3)$. We consider an example of a relation algebra $\mathbf{A}$, whose NSP can be solved by a $(2,3)$-Datalog program.

5.1 Example Let $A_0 = \{0,1,2,3,4,5\}$ be the atoms of a relation algebra $\mathbf{A}$ and the multiplication table for the atoms is given in Figure 5.1. Note that this multiplication table encodes exactly the allowed triples of triangle inequalities on the distance set $\{0,1,2,3,4,5\}$, i.e., for example $1 \circ 3 = \{2,3,4\}$, since $(1,3,2)$, $(1,3,3)$, and $(1,3,4)$ satisfy all instantiations of triangle inequalities and $(1,3,0)$, $(1,3,1)$, and $(1,3,5)$ do not (cf. Example 4.23). We denote the elements of $\mathbf{A}$ by subsets of the set of atoms A_0. We want to mention that $\mathbf{A}$ has a normal representation according to Lemma 3.1.1 in [PR96] (see also Proposition 2.7.4. in [Con15]).

Consider a copy $\mathbf{A}^*$ of the relation angebra $\mathbf{A}$ that behaves exactly in the same way as $\mathbf{A}$, i.e., $a^* \circ b^* = c^*$ holds for $a^*, b^*, c^* \in A^*$ if and only if $a \circ b = c$ holds for $a, b, c \in A$. Analogously to $\mathbf{A}$ the elements of $\mathbf{A}^*$ are denoted by subsets of $A_0^* = \{0^*, \dots, 5^*\}$. We use the elements of $\mathbf{A}^*$ as IDBs in the following $(2,3)$-Datalog program Π that solves $\mathrm{NSP}(\mathbf{A})$.

$$
\begin{array}{lr}
\text{For every } \{a,b,c,d\} \subseteq \{0,\dots,5\}: & \{a^*,b^*,c^*,d^*\}(x,y) :- \{a,b,c,d\}(x,y) \\
\text{For every } A, B \subseteq \{0^*,\dots,5^*\}: & (A \cap B)(x,y) :- A(x,y), B(x,y) \\
\text{For every } A, B \subseteq \{0^*,\dots,5^*\}: & (A \circ B)(x,y) :- A(x,z), B(z,y) \\
 & \mathit{false} :- \{\}(x,y)
\end{array}
$$

The intuition of this program is that it checks for a given instance network $(V; f)$ for all elements x, y, z in the domain of f and every atom $a \leqslant f(x,y)$ whether there exist atoms $b \leqslant f(x,z)$ and $c \leqslant f(y,z)$ such that (a,b,c) is an allowed triple, i.e., $a \leqslant b \circ c$ holds.

$\circ$	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	$\{0,1,2\}$	$\{1,2,3\}$	$\{2,3,4\}$	$\{3,4,5\}$	$\{4,5\}$
2	2	$\{1,2,3\}$	$\{0,1,2,3,4\}$	$\{1,2,3,4,5\}$	$\{2,3,4,5\}$	$\{3,4,5\}$
3	3	$\{2,3,4\}$	$\{1,2,3,4,5\}$	$\{0,1,2,3,4,5\}$	$\{1,2,3,4,5\}$	$\{2,3,4,5\}$
4	3	$\{3,4,5\}$	$\{2,3,4,5\}$	$\{1,2,3,4,5\}$	$\{0,1,2,3,4,5\}$	$\{1,2,3,4,5\}$
5	3	$\{4,5\}$	$\{3,4,5\}$	$\{2,3,4,5\}$	$\{1,2,3,4,5\}$	$\{0,1,2,3,4,5\}$

Figure 5.1: Multiplication table of a relation algebra.

If this is not the case a is "removed" from $f(x,y)$. This is done by adding (x,y) to the relation $f(x,z) \circ f(z,y)$. Since we cannot change the EDBs, we do all this with the copies of the relations. The program Π continues until nothing changes. If there exists one edge (x,y) where $\{\}(x,y)$ is derived, then the instance is rejected, otherwise it is accepted. It is clear that every network that is rejected by Π is not satisfiable. For an accepted instance we choose for every edge (x,y) the maximal distance from $\{0^*,\dots,5^*\}$ which is present in all predicates $A \subseteq \{0^*,\dots,5^*\}$ that hold for (x,y) in $\Pi((V;f))$. In this way we get an atomic network. With the help of the triangle inequalities it can be checked that this atomic network is even closed and therefore satisfied in the normal representation of $\mathbf{A}$. The Datalog program Π solves NSP($\mathbf{A}$).

One reason why Datalog programs of width $(2,3)$ are of central interest in the study of network satisfaction problems is that they can be formulated purely in terms of the relation algebra (see, e.g., [BJ17]). In contrast to other consistency notions, such as Datalog programs of arbitrary width, it is not necessary to consider an underlying representation of the relation algebra. The reader may notice in this context how compact the presentation of the $(2,3)$-Datalog program from the previous example was. For a Datalog program of arbitrary width, we would have to introduce higher-order relation symbols that have no equivalents in the elements of the relation algebra.

It would be of great interest to have a clear picture of which polynomial-time tractable NSPs are also solvable by a Datalog program of width $(2,3)$. However, it is not even known which tractable NSPs can be solved by a Datalog program of arbitrary width. In this chapter, we answer this second question for NSPs of finite symmetric relation algebras with a flexible atom. In fact, we prove something more general.

5.2 Theorem (Datalog Tractability) *Let $\mathbf{A}$ be a finite symmetric relation algebra with a normal representation $\mathfrak{B}$ that has an injective polymorphism, and let A_0 be the set of atoms of $\mathbf{A}$. If there exists an operation $f\colon A_0^6 \to A_0$ such that*

1. *f preserves the allowed triples of $\mathbf{A}$,*

2. $\forall x_1, \ldots, x_6 \in A_0.\, f(x_1, \ldots, x_6) \in \{x_1, \ldots x_6\}$,

3. *f satisfies the Siggers identity:* $\forall x, y, z \in A_0.\, f(x, x, y, y, z, z) = f(y, z, x, z, x, y)$,

then the network satisfaction problem for **A** *can be solved by a Datalog program.*

Note that this theorem does not make the assumption that the relation algebra has a flexible atom. However, with the main result of the previous chapter the following characterization of Datalog solvability for NSPs of finite symmetric integral relation algebras with a flexible atom follows immediately.

5.3 Theorem *Let* **A** *be a finite symmetric integral relation algebra with a flexible atom and let* A_0 *be the set of atoms of* **A**. *Then the following are equivalent:*

- *There exists an operation* $f\colon A_0^6 \to A_0$ *such that*
  1. *f preserves the allowed triples of* **A**,
  2. $\forall x_1, \ldots, x_6 \in A_0.\, f(x_1, \ldots, x_6) \in \{x_1, \ldots x_6\}$,
  3. *f satisfies the Siggers identity:* $\forall x, y, z \in A_0.\, f(x, x, y, y, z, z) = f(y, z, x, z, x, y)$,
- NSP(**A**) *can be solved by a Datalog program.*

Proof: The implication from the first item to the second item follows directly from Proposition 4.24 and Theorem 5.2. We prove the second implication by showing the contraposition. Assume that the first item is not satisfied. Let $\mathfrak{B}$ be the normal representation of **A**. Then Theorem 4.45 implies that $\mathrm{HSP}^{\mathrm{fin}}(\{\mathrm{Pol}(\mathfrak{B})\})$ contains a 2-element algebra where all operations are projections. By Corollary 2.47 the clone $\mathrm{Pol}(\mathfrak{B})$ has a uniformly continuous clone homomorphism to Proj. Since Proj is contained in Pol(LIN), this implies that there exists also a uniformly continuous clone homomorphism from $\mathrm{Pol}(\mathfrak{B})$ to Pol(LIN). By Corollary 2.58 we get that $\mathrm{CSP}(\mathfrak{B})$ and therefore NSP(**A**) cannot be solved by a Datalog program. This proves the second implication. □

As another consequence of Theorem 5.2 we get the following strengthening of the complexity dichotomy NSPs of finite symmetric integral relation algebra with a flexible atom.

5.4 Corollary (Complexity Dichotomy) *For every finite symmetric integral relation algebra* **A** *with a flexible atom,* NSP(**A**) *can be solved by a Datalog program or is NP-complete.*

Proof: Suppose the first condition in Theorem 4.45 holds. It follows from Proposition 4.24 that the normal representation $\mathfrak{B}$ of **A** has an injective polymorphism and therefore Theorem 5.2 applies and yields the result. □

5.1.1 Outline of the Proof and Preliminaries

We will outline the proof of Theorem 5.2 in this section and cite some results from the literature that we will use later. Assume that $\mathbf{A}$ is a finite symmetric relation algebra that satisfies the assumptions of Theorem 5.2. From the results in Section 4.3 it follows that the atom structure of $\mathbf{A}$, denoted by $\mathfrak{A}_0$, has a polymorphism that is a Siggers operation. Moreover, $\mathfrak{A}_0$ is a *conservative structure*, i.e., $\mathrm{Pol}(\mathfrak{A}_0)$ is a conservative polymorphism clone.

Recall the notion of semilattice, majority, and affine edges for conservative structures (cf. Definition 2.41). The existence of a Siggers operation in $\mathrm{Pol}(\mathfrak{A}_0)$ ensures by Theorem 2.43 that every edge in $\mathfrak{A}_0$ is semilattice, majority, or affine. In Section 5.2.1 we analyze the different types of edges in the atom structure $\mathfrak{A}_0$ and obtain results about their appearance. Our goal is to show that there are no affine edges in $\mathfrak{A}_0$, since we will see that this implies solvability of a Datalog program. Fortunately, there is the following result by Alexandr Kazda about *binary* conservative structures, i.e., structure where all relations have arity at most two.

5.5 Theorem (Theorem 4.5 in [Kaz15]) *If $\mathfrak{A}$ is a finite conservative binary relational structure with a Siggers polymorphism, then $\mathfrak{A}$ has no affine edges.*

You will have noticed 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 $\mathfrak{A}_0$". In Section 5.2.2 we give the formal definition of $\mathfrak{A}_0^{\mathrm{b}}$ and 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 $\mathfrak{A}_0$ also has no affine edge. This is done in Section 5.2.3.

Having shown that $\mathfrak{A}_0$ has no affine edge, we use the fact that this implies Datalog solvability for $\mathrm{CSP}(\mathfrak{A}_0)$, which was obtained in [Bul11]. We present this fact here via a characterization of Datalog solvability for $\mathrm{CSP}(\mathfrak{A}_0)$ in terms of weak near-unanimity polymorphisms from Theorem 2.56.

5.6 Proposition (cf. Corollary 3.2 in [Kaz15]) *If $\mathfrak{A}_0$ is 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*

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

Now we have to take the last step and justify that the Datalog solvability of $\mathrm{CSP}(\mathfrak{A}_0)$ (or its characterization in terms of polymorphims from the previous proposition) leads to the Datalog solvability of $\mathrm{CSP}(\mathfrak{B})$ for the normal representation $\mathfrak{B}$ of $\mathbf{A}$ and thus to the Datalog solvability of $\mathrm{NSP}(\mathbf{A})$.

For this reduction we use the following theorem from [BM16]. We present it here in a specific formulation that already incorporates our knowledge about canonical functions and their correspondence to polymorphisms of the atom structure from Section 4.3.

5.7 Theorem (Theorem 8 in [BM16]) *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 $\mathrm{CSP}(\mathfrak{B})$ *is in Datalog.*

This concludes the outline of the proof. In Section 5.2.4 we put things together and conclude the proof of Theorem 5.2. We will finish Chapter 5 in Section 5.3 with examples and a discussion of the result.

5.2 Proof of the Datalog Tractability Theorem

5.2.1 The Atom Structure

To keep the presentation clear, we will make some global assumptions for Sections 5.2.1-5.2.3. In the following, let **A** be a finite relation algebra that satisfies the assumptions from Theorem 5.2: **A** has a normal representation $\mathfrak{B}$ with an injective polymorphism and there exists an operation $f' \colon A_0^6 \to A_0$ such that

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

Furthermore, let $\mathfrak{A}_0$ be the atom structure of **A**. Recall the definition of the atom structure in Section 4.3.

5.8 Definition The atom structure $\mathfrak{A}_0$ has as its domain the set of atoms A_0 of **A** and it has the following relations:

- unary relation U_S for every subset S of A_0;
- a ternary relation $R \subseteq A^3$ that consists of the allowed triples of **A**.

5.9 Remark Since **A** is a symmetric relation algebra, the relation R is totally symmetric. Furthermore, we can drop the binary relation E (cf. Definition 4.12), since it consists only of loops and does not change the set of polymorphisms.

The assumptions on **A** give us even more: Since **A** satisfies the assumption from Theorem 5.2 it follows by the results of Section 4.3 that $\mathrm{Pol}(\mathfrak{A}_0)$ contains a Siggers operation. Recall that this implies by Theorem 2.43 that for every $a, b \in A_0$ the set $\{a, b\}$ is a majority edge, an affine edge, or there is a semilattice edge on $\{a, b\}$. The different types of edges are witnessed by certain operations that we get from Proposition 2.42: 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 whenever 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.

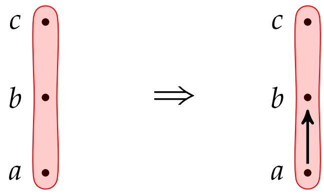


Figure 5.2: The statement of Lemma 5.11. The red shape means $(a, b, c) \notin R$, the black arrow means $(a, a, b) \notin R$.

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.

The following series of lemmas gives us insights into the constitution of the relation R.

5.10 Lemma *The relation R of the atom structure $\mathfrak{A}_0$ has the following properties:*

- *for all $a, b \in A_0$ we have $(a, a, b) \in R$ or $(a, b, b) \in R$;*
- *for all $a \in A_0$ we have $(a, a, a) \in R$.*

Proof: Let $\mathfrak{B}$ be the normal representation of $\mathbf{A}$ and recall that $\mathfrak{B}$ has an injective polymorphism i by the assumptions of Theorem 5.2. Let $x_1, x_2, y_1, y_2 \in B$ be such that $a^{\mathfrak{B}}(x_1, x_2)$ and $b^{\mathfrak{B}}(y_1, y_2)$. 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, b) or (a, b, b) are allowed triples and therefore $(a, a, b) \in R$ or $(a, b, b) \in R$.

For the second claim 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 and therefore $(a, a, a) \in R$. □

5.11 Lemma *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.

5.12 Lemma *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 5.10 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 the fact 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. □

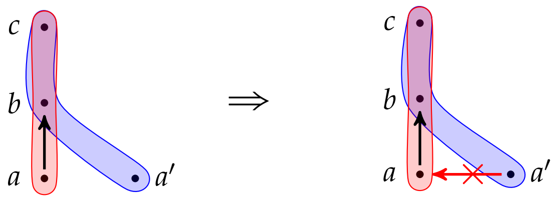


Figure 5.3: The statement of Lemma 5.13. The blue shape means $(a', b, c) \in R$, the crossed-out red arrow means (a', a) is not a semilattice edge.

5.13 Lemma *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 5.10 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.

This proves Claim 2.

Claim 3: $p(c, a) = a$ implies $p(a, c) = c$. Lemma 5.10 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 $p((a,a,a),(a',b,c)) = (a,b,c) \in R$.

This proves the lemma. $\square$

5.2.2 The Binarisation

We have announced in the introduction that we want to apply Kazda's theorem (Theorem 5.5) for binary conservative structures, but the atom structure $\mathfrak{A}_0$ from Section 4.3 has a ternary relation. We therefore associate a certain binary structure $\mathfrak{A}_0^{\mathrm{b}}$ to $\mathfrak{A}_0$ which shares many properties with $\mathfrak{A}_0$. (Recall that the fixed atom structure $\mathfrak{A}_0$ of the relation algebra $\mathbf{A}$ has all unary relations and the ternary relation R as relations.)

5.14 Definition We denote by $\mathfrak{A}_0^{\mathrm{b}}$ the structure with domain A_0 that contains the following relations:

- a unary relation U_S for each subset S of A_0;
- for every $a \in A_0$ the binary relation $R_a := \{(x,y) \in A_0^2 \mid (a,x,y) \in R\}$;
- arbitrary unions of relations of the form R_a.

In the following $\mathfrak{A}_0^{\mathrm{b}}$ denotes the binarisation of $\mathfrak{A}_0$ according to Definition 5.14. We obtain the following results about the relationship of $\mathrm{Pol}(\mathfrak{A}_0)$ and $\mathrm{Pol}(\mathfrak{A}_0^{\mathrm{b}})$.

5.15 Lemma $\mathrm{Pol}(\mathfrak{A}_0) \subseteq \mathrm{Pol}(\mathfrak{A}_0^{b})$.

Proof: Clearly, each relation R_a has the primitive positive definition $\exists z(U_{\{a\}}(z) \wedge R(z,x,y))$ in $\mathfrak{A}_0$. A primitive positive definition of $\cup_{a \in S} R_a$ is $\exists z(U_S(z) \wedge R(z,x,y))$ in $\mathfrak{A}_0$. Then the statement of the lemma follows by Theorem 2.32. $\square$

5.16 Lemma $\mathrm{Pol}^{(2)}(\mathfrak{A}_0^{b}) \subseteq \mathrm{Pol}^{(2)}(\mathfrak{A}_0)$.

Proof: Let $f \in \mathrm{Pol}^{(2)}(\mathfrak{A}_0^{\mathrm{b}})$. It suffices to prove that the operation f preserves the relation R. Let $(a_1,b_1,c_1),(a_2,b_2,c_2) \in R$ be arbitrary. We want to show that the triple $t := (f(a_1,a_2), f(b_1,b_2), f(c_1,c_2))$ is in R as well. If $t \in \{(a_1,b_1,c_1),(a_2,b_2,c_2)\}$ then there is nothing to be shown. Otherwise, since f must preserve $\{a_1,a_2\}$, $\{b_1,b_2\}$, and $\{c_1,c_2\}$, by the symmetry of R and possibly flipping the arguments of f we may assume without loss of generality that $f(a_1,a_2) = a_1$, $f(b_1,b_2) = b_1$, and $f(c_1,c_2) = c_2$. So we have to show that $t = (a_1,b_1,c_2) \in R$. Note that $(b_1,c_1) \in R_{a_1}$ and $(b_2,c_2) \in R_{a_2}$, and therefore $(f(b_1,b_2), f(c_1,c_2)) \in R_{a_1} \cup R_{a_2}$. In the first case, we obtain that $(b_1,c_2) \in R_{a_1}$, and hence $(a_1,b_1,c_2) \in R$ and we are done. In the second case, we obtain that $(b_1,c_2) \in R_{a_2}$, and hence $(a_2,b_1,c_2) \in R$. In partciular, $(a_2,c_2) \in R_{b_1}$. Since $(a_1,c_1) \in R_{b_1}$ and since f preserves R_{b_1} we have that $(f(a_1,a_2), f(c_1,c_2)) = (a_1,c_2) \in R_{b_1}$, and hence $(a_1,b_1,c_2) \in R$, which concludes the proof. $\square$

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

Figure 5.4: Multiplication table of the relation algebra $\mathbf{K}$.

We want to remark that this implies that $\mathfrak{A}_0^{\mathrm{b}}$ and $\mathfrak{A}$ have exactly the same semilatice edges. The following example shows that in general it does not hold that $\mathrm{Pol}(\mathfrak{A}_0^{\mathrm{b}}) \subseteq \mathrm{Pol}(\mathfrak{A}_0)$.

5.17 Example Let $\mathbf{K}$ be the relation algebra with two atoms $\{\mathrm{Id}, E\}$ and the multiplication table given in Figure 5.4. It is easy to see that the expansion of the infinite clique K_ω by all first-order definable binary relations is a normal representation of $\mathbf{K}$. Then $\mathfrak{K}_0$ does not have a majority polymorphism, but $\mathfrak{K}_0^b$ does since every binary relation on a two-element set is preserved by the (unique) majority operation on a two-element set.

5.2.3 No Affine Edges in the Atom Structure

We show in this section that under the assumption that $\mathfrak{A}_0^{\mathrm{b}}$ has a Siggers polymorphism and has no affine edge, $\mathfrak{A}_0$ also has no affine edge. So let us assume for the whole section that $\mathrm{Pol}(\mathfrak{A}_0^{\mathrm{b}})$ contains a Siggers operation and that $\mathfrak{A}_0^{\mathrm{b}}$ has no affine edge.

Since $\mathfrak{A}_0^{\mathrm{b}}$ is conservative and has no affine edge, there exists according to Proposition 2.42 a binary operation $v \in \mathrm{Pol}(\mathfrak{A}_0)$ and a ternary operation $w \in \mathrm{Pol}(\mathfrak{A}_0)$ such that for every two element subset C of A_0,

- $v|_C$ is a semilattice operation whenever C has a semilattice edge, and $v|_C(x, y) = x$ otherwise;
- $w|_C$ is a majority operation if C is a majority edge and $w|_C(x, y, z) = v|_C(v|_C(x, y), z)$ if C has a semilattice edge.

We define

$$u(x, y, z) := w(v(v(x, y), z), v(v(y, z), x), v(v(z, x), y)).$$

5.18 Lemma *The structures $\mathfrak{A}_0$ and $\mathfrak{A}_0^{\mathrm{b}}$ have exactly the same semilattice edges. Let $a, b \in A_0$ be such that $\{a, b\}$ has no semilattice edge in one of the structures. Then the restriction of u to $\{a, b\}$ is a majority operation.*

Proof: By Lemma 5.15 and Lemma 5.16, the structures $\mathfrak{A}_0$ and $\mathfrak{A}_0^{\mathrm{b}}$ have exactly the same semilattice edges, since they have the same binary polymorphisms. The second statement follows from the definition of u by means of w and v. □

5.19 Definition Let f be a binary operation on A_0. Then we say that $\{a,b,c\} \subseteq A_0$ has the *f-cycle* (x,y,z) if $\{x,y,z\} = \{a,b,c\}$ and (x,y), (y,z), and (z,x) are f-sl.

5.20 Lemma *Let $a,b,c \in A_0$ be such that (a,b) is v-sl but (a,b,c) is not a v-cycle. Then $u(r,s,t) \neq a$ for any choice of $r,s,t \in A_0$ such that $\{r,s,t\} = \{a,b,c\}$.*

Proof: We prove a series of intermediate claims.

Claim 1: If $\{x,y,z\} = \{a,b,c\}$ and $v(v(x,y),z) = a$, then $z = a$. We assume for contradiction that $z \neq a$ and distinguish the following cases.

1. $x = a, y = b, z = c$: Then $v(v(x,y),z) = v(b,c) \in \{b,c\}$.
2. $x = a, y = c, z = b$: Then $v(v(x,y),z) \in \{v(a,b), v(c,b)\} \subseteq \{b,c\}$.
3. $x = b, y = a, z = c$: Then $v(v(x,y),z) = v(b,c) \in \{b,c\}$.
4. $x = c, y = a, z = b$: Then $v(v(x,y),z) \in \{v(c,b), v(a,b)\} \subseteq \{b,c\}$.

In all four cases we have $v(v(x,y),z) \neq a$, which contradicts our assumption and proves the claim.

Claim 2: If $\{x,y,z\} = \{a,b,c\}$ and $v(v(x,y),z) = a$, then (c,a) is v-sl.
By Claim 1 we get that $z = a$ and furthermore we have $v(x,y) = v(b,c) = c$ since otherwise $v(x,y) = v(b,c) = b$ and $v(v(x,y),z) = v(b,a) = b$, which contradicts our assumption. Assume for contradiction that (c,a) is not v-sl and therefore one of the following holds:

1. (a,c) is v-sl. It follows that $v(v(x,y),z) = v(c,a) = c$ which contradicts our assumption.
2. $\{a,c\}$ is a majority edge of $\mathfrak{A}_0^{\mathrm{b}}$. It follows again that $v(v(x,y),z) = v(c,a) = c$, since v behaves like the projection on the first coordinate on majority edges. This contradicts our assumption.

Claim 3: If $\{x,y,z\} = \{a,b,c\}$ and $v(v(x,y),z) = a$, then $\{b,c\}$ is a majority edge of $\mathfrak{A}_0^{\mathrm{b}}$.
Assume for contradiction that there is a semilattice edge on $\{b,c\} = \{x,y\}$. By Claim 2 and our assumption that (a,b,c) is not a v-cycle, the edge (b,c) is not v-sl and therefore (c,b) is v-sl. Therefore, we get $v(v(x,y),z) = v(b,a) = b$ which contradicts our assumption.

Claim 4: If $\{x,y,z\} = \{a,b,c\}$ and $v(v(x,y),z) = a$, then $v(v(z,x),y) = b = v(v(y,z),x)$ follows. By Claim 3, $\{b,c\}$ is a majority edge of $\mathfrak{A}_0^{\mathrm{b}}$ and it follows that $b = y$ and $c = x$ since otherwise $v(v(x,y),z) = v(b,a) = b$. Now we calculate

$$v(v(z,x),y)) = v(v(a,c),b)) = v(a,b) = b = v(b,c) = v(v(b,a),c) = v(v(y,z),x)$$

which proves the claim.

Now we are able to prove the statement of the lemma. Assume for contradiction that $u(r,s,t) = a$. Since w preserves $U_{A\setminus\{a\}}$ this is only possible if at least one of the terms $v(v(r,s),t), v(v(s,t),r)$, or $v(v(t,r),s))$ evaluates to a. By Claim 4 we get that the two other terms evaluate to b. Since (a,b) is v-sl we get that $w(a,b,b) = w(b,a,b) = w(b,b,a) = b$ which contradicts our assumption $u(r,s,t) = a$. □

5.21 Theorem *If $u \in \mathrm{Pol}(\mathfrak{A}_0^b)$, then $u \in \mathrm{Pol}(\mathfrak{A}_0)$.*

Proof: We have to prove that u preserves R. Let $(a_1,b_1,c_1),(a_2,b_2,c_2),(a_3,b_3,c_3) \in R$ and let

$$(a,b,c) := (u(a_1,a_2,a_3), u(b_1,b_2,b_3), u(c_1,c_2,c_3)).$$

Assume for contradiction that $(a,b,c) \notin R$. By Lemma 5.11, we may assume without loss of generality that $(a,a,b) \notin R$ and hence by Lemma 5.12 (a,b) is a semilattice edge in $\mathfrak{A}_0$ and (b,a) is not. By Lemma 5.18 the structures $\mathfrak{A}_0$ and $\mathfrak{A}_0^{\mathrm{b}}$ have exactly the same semilattice edges. This implies that $\{a,b\}$ has a semilattice edge; this semilattice edge can only be (a,b) and therefore (a,b) is v-sl. Since u preserves $R_{a_1} \cup R_{a_2} \cup R_{a_3}$ there exists $r \in \{a_1,a_2,a_3\}$ such that $(r,b,c) \in R$. By Lemma 5.13 we get that (r,a) is not a semilattice edge in $\mathfrak{A}_0$ and therefore Lemma 5.16 implies that (r,a) is not a semilattice edge in $\mathfrak{A}_0^{\mathrm{b}}$ and we get that (r,a) is not v-sl. Let $s \in \{a_1,a_2,a_3\}\setminus\{a,r\}$.

<u>Claim1:</u> $\{a,r,s\}$ does not have a v-cycle. Assume for contradiction that $\{a,r,s\}$ has a v-cycle. Since (r,a) is not v-sl it follows that (a,r) is v-sl and therefore (s,a) is v-sl. We consider the following two cases:

1. $(s,b,c) \in R$. Then Lemma 5.13 applied to a,s,b,c implies that (s,a) is not a semilattice edge and therefore by Lemma 5.16 (s,a) is not v-sl, which is a contradiction.
2. $(s,b,c) \notin R$. Note that $(s,s,b) \notin R$ holds, since (s,a) is v-sl and $v((s,s,b),(a,a,a)) = (a,a,b) \in R$ yields a contradiction to $(a,a,b) \notin R$. Hence, Lemma 5.13 applied to s,r,b,c implies that (r,s) is not a semilattice edge and therefore by Lemma 5.16 (r,s) is not v-sl, which is again a contradiction.

This proves that $\{a,r,s\}$ cannot have a v-cycle.

<u>Claim 2:</u> $u(a,a,r) = a$. Assume for contradiction that $u(a,a,r) = r$. Then $\{a,r\}$ is clearly not a majority edge of $\mathfrak{A}_0^{\mathrm{b}}$, and since $\mathfrak{A}_0^{\mathrm{b}}$ does not have affine edges it follows that (a,r) is v-sl. Furthermore, (a,r,s) is not a v-cycle and therefore Lemma 5.20 implies that $u(a_1,a_2,a_3) \neq a$ which contradicts the definition of a.

Finally, consider the following application of the polymorphism u:

$$u\big((a,a),(a,a),(r,b)\big) = (a,b).$$

Since $(a,a) \in R_a$ and $(r,b) \in R_c$ and since u is in $\mathrm{Pol}(\mathfrak{A}_0^b)$ we get that $(a,b) \in R_a \cup R_c$. Hence, $(a,a,b) \in R$ or $(c,a,b) \in R$, which contradicts our assumptions. □

5.2.4 Final Proof of the Main Result

We can now prove the main result of this chapter.

Proof of Theorem 5.2:
Let $\mathbf{A}$ be a finite relation algebra that satisfies the assumptions and let $\mathfrak{A}_0$ be the atom structure of $\mathbf{A}$. We denote by $\mathfrak{A}_0^{\mathrm{b}}$ the binarisation of $\mathfrak{A}_0$ according to Definition 5.14. It follows from the assumptions on $\mathbf{A}$ that $\mathrm{Pol}(\mathfrak{A}_0)$ contains a Siggers operation. By Lemma 5.15 we get that $\mathrm{Pol}(\mathfrak{A}_0^{\mathrm{b}})$ contains a Siggers operation as well. Note that $\mathfrak{A}_0^{\mathrm{b}}$ is a finite binary conservative structure and therefore Theorem 5.5 implies that $\mathfrak{A}_0^{\mathrm{b}}$ has no affine edges. Therefore, $\mathfrak{A}_0^{\mathrm{b}}$ satisfies the general assumption from Section 5.2.3 and we can define the operation u as it is done in the beginning of this section. Note that u witnesses by Lemma 5.18 that $\mathfrak{A}_0^{\mathrm{b}}$ does not have an affine edge. We can now apply Theorem 5.21 and get that u is also a polymorphism of $\mathfrak{A}_0$. Recall that $\mathfrak{A}_0$ and $\mathfrak{A}_0^{\mathrm{b}}$ have by Lemma 5.16 exactly the same semilattice edges and therefore Lemma 5.18 and the fact that u is a polymorphism of $\mathfrak{A}_0$ imply that $\mathfrak{A}_0$ does not have an affine edge. By Proposition 5.6 we get that there exists a 3-ary weak near unanimity polymorphism $f \in \mathrm{Pol}(\mathfrak{A}_0)$ and a 4-ary weak near unanimity polymorphism $g \in \mathrm{Pol}(\mathfrak{A}_0)$ such that

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

holds. Theorem 5.7 implies that $\mathrm{CSP}(\mathfrak{B})$ and thus also $\mathrm{NSP}(\mathbf{A})$ can be solved by a Datalog program. $\square$

$\circ$	Id	$<$	$>$	$\vert$
Id	Id	$<$	$>$	$\vert$
$<$	$<$	$<$	$\{\mathrm{Id}, <, >\}$	$\{<, \vert\}$
$>$	$>$	1	$>$	$\vert$
$\vert$	$\vert$	$\vert$	$\{>, \vert\}$	1

Figure 5.5: Multiplication table of the left-linear point algebra **LLP**.

5.3 Examples and Discussion

As mentioned in the introduction, an important goal in the study of NSPs is to understand which problems can be solved by a Datalog program, in particular by a $(2, 3)$-Datalog program.

Question *Which finite relation algebras have a NSP that*

- *can be solved by a Datalog program?*
- *can be solved by a Datalog program of width* $(2, 3)$*?*

We would like to give a brief overview of the state of research, formulate some open problems and discuss our findings. Let us first look at how these questions relate to Hirsch's RBCP. A priori, it could be that every polynomial-time solvable NSP is also solvable by a Datalog program of width $(2, 3)$. However, there is an example of a relation algebra whose NSP is in P but cannot be solved by a Datalog program of any width.

5.22 Example The *left-linear point algebra* **LLP** (see [Hir97, Dün05]) consists of the atoms $LLP_0 = \{\mathrm{Id}, <, >, |\}$ and the composition operation is given by the multiplication table in Figure 5.5. There exist several polynomial-time algorithms that solve the network satisfaction problem of **LLP** [Hir97, BJ03, BK02]. On the other hand, NSP(**LLP**) cannot be solved by a Datalog program of any width (see Problem 3.8 in [BM11] and consider [Bod21] for the proof). We would like to note that **LLP** has a fully universal square representation (Example 5.5.6 in [Bod21]).

As in the case of the RBCP, it might be promising to restrict the Datalog classification task to relation algebras with a normal representation. How is the situation for those relation algebras? The left-linear point algebra has no normal representation (Example 7 in [Bod18]), and we unfortunately lack an example of a polynomial-time tractable NSP for a relation algebra with normal representation that cannot be solved by a Datalog program.

5.23 Problem *Find a finite relation algebra with a normal representation that has a polynomial-time tractable NSP which cannot be solved by a Datalog program .*

To our knowledge, even the following problem is open.

5.24 Problem *Find a finite relation algebra with a normal representation that has a polynomial-time tractable NSP which cannot be solved by a Datalog program of width* $(2,3)$.

The main result of this section is that there is no such example in the class of finite symmetric relation algebras with a flexible atom. However, we do not know whether all these problems can be solved by a program of width $(2,3)$. A close inspection of the reduction to the CSP of the atom structure (Section 4.3) yields width $(4,6)$. It might be possible to do a little better (see for example [MNPW21]) but we do not see how width $(2,3)$ can be shown. We are optimistic that an analysis analogous to the recent work [Wro20a, Wro20b] can achieve a solution to the following problem.

5.25 Problem *Prove Theorem 5.2 with solvability by a Datalog program of width* $(2,3)$ *in the conclusion.*

As we have seen, the proof of Theorem 5.2 is based on the Datalog solvability of the CSP of the atom structure. This procedure cannot be transferred to relation algebras with normal representations. There are several counterexamples at different levels. The first example is the point algebra.

5.26 Example Recall the point algebra from Example 2.5. This relation algebra has a normal representation and its NSP is solvable by a $(2,3)$-Datalog program. However, $\mathrm{CSP}(\mathfrak{A}_0)$ is NP-hard for its atom structure $\mathfrak{A}_0$ and does not serve as a reason for the polynomial-time solvability of the NSP, let alone for its Datalog solvability.

This example raises the question whether, at least under the assumption that $\mathrm{CSP}(\mathfrak{A}_0)$ is in P, the Datalog solvability of the NSP is always determined by the Datalog solvability of $\mathrm{CSP}(\mathfrak{A}_0)$. Surprisingly, this is not the case either, as the following example shows.

5.27 Example Consider the relation algebra $\mathbf{C}$ with atoms $\{\mathrm{Id}, E, N\}$ and the multiplication table in Figure 5.6. 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 $\mathrm{CSP}(\overline{\omega K_2})$ and therefore also the NSP of the relation algebra can be solved by the following $(2,3)$-Datalog program. Note that we assume that the relations of the instance are symmetric in order to keep the presentation of the following program clear.

$$
\begin{aligned}
D(x,y) &:- \; E(x,y) \\
D(x,y) &:- \; (\mathrm{Id} \cup E)(x,y) \\
\textit{false} &:- \; D(x,y), N(x,y) \\
\mathrm{Id}^*(x,y) &:- \; (\mathrm{Id} \cup N)(x,y), D(x,y)
\end{aligned}
\qquad
\begin{aligned}
D(x,y) &:- \; \mathrm{Id}(x,y) \\
D(x,y) &:- \; D(x,z), D(z,y)
\end{aligned}
$$

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

Figure 5.6: Multiplication table of the relation algebra **C**.

$$\begin{aligned} E^*(x,y) &:- (E \cup N)(x,y), D(x,y) && \\ \mathrm{Id}^*(x,y) &:- \mathrm{Id}(x,y) & \mathrm{Id}^*(x,x) &:- \\ \mathrm{Id}^*(x,y) &:- \mathrm{Id}^*(x,z), \mathrm{Id}^*(z,y) & \mathrm{Id}^*(x,y) &:- E^*(x,z), E^*(y,z) \\ E^*(x,y) &:- E(x,y) & E^*(x,y) &:- E^*(x,z), \mathrm{Id}^*(y,z) \\ \mathit{false} &:- \mathrm{Id}^*(x,y), E^*(x,y) && \end{aligned}$$

One can check that this program solves $\mathrm{CSP}(\overline{\omega K_2})$. Let $\mathfrak{C}_0$ be the atom structure of the relation algebra. It follows from Proposition 33 in [BMPP19] that $\mathfrak{C}_0$ has a Siggers polymorphism and therefore $\mathrm{CSP}(\mathfrak{C}_0)$ is polynomial-time solvable. However, $\mathrm{CSP}(\mathfrak{C}_0)$ is not solvable by a Datalog program. We argue that $\mathfrak{C}_0$ has an affine edge on $\{\mathrm{Id}, E\}$ which implies by [Bul11] that $\mathrm{CSP}(\mathfrak{C}_0)$ cannot be solved by a Datalog program. Suppose for contradiction that $\mathfrak{C}_0$ has a semilattice edge on $\{\mathrm{Id}, E\}$, witnessed by a polymophism f. By the definition of the relation algebra the tuples (E,E,E) and $(\mathrm{Id},\mathrm{Id},E)$ are not in the relation $R^{\mathfrak{C}_0}$, but (E,E,Id) is. This yields a contradiction for $f((E,E,\mathrm{Id}),(E,\mathrm{Id},E))$ for both possibilities $f(E,\mathrm{Id}) = E = f(\mathrm{Id},E)$ and $f(E,\mathrm{Id}) = \mathrm{Id} = f(\mathrm{Id},E)$. Furthermore, there can also be no majority operation m on $\{\mathrm{Id}, E\}$, since $m((E,E,\mathrm{Id}),(E,\mathrm{Id},E),(\mathrm{Id},E,E,)) = (E,E,E) \in R^{\mathfrak{C}_0}$ is a contradiction. Since $\mathfrak{C}_0$ has a Siggers polymorphism, $\{\mathrm{Id}, E\}$ is an affine edge.

This example indicates that the Datalog classification question for NSPs of relation algebras with a normal representation cannot be shown by means of polymorphisms of the atom structure, or, in other words, by means of canonical polymorphisms. We want to point out recent results in [MNPW21] and [BR22] concerning these questions in other classes of reducts of finitely bounded homogeneous structures.

Bibliography

[AB09] Sanjeev Arora and Boaz Barak, *Computational complexity: A modern approach*, 1st ed., Cambridge University Press, USA, 2009.

[ABD09] Albert Atserias, Andrei A. Bulatov, and Anuj Dawar, *Affine systems of equations and counting infinitary logic*, Theoretical Computer Science **410** (2009), no. 18, 1666–1683.

[All83] James F. Allen, *Maintaining knowledge about temporal intervals*, Communications of the ACM **26** (1983), no. 11, 832–843.

[AM94] Hajnal Andréka and Roger D. Maddux, *Representations for small relation algebras*, Notre Dame Journal of Formal Logic **35** (1994), no. 4, 550–562.

[AMM08] Jeremy F. Alm, Roger D. Maddux, and Jacob Manske, *Chromatic graphs, Ramsey numbers and the flexible atom conjecture*, The Electronic Journal of Combinatorics **15** (2008), no. 1.

[Bar11] Libor Barto, *The dichotomy for conservative constraint satisfaction problems revisited*, Proceedings of the Symposium on Logic in Computer Science (LICS) (Toronto, Canada), 2011.

[BB21] Manuel Bodirsky and Bertalan Bodor, *Canonical polymorphisms of ramsey structures and the unique interpolation property*, 2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 2021, pp. 1–13.

[BD13] Manuel Bodirsky and Víctor Dalmau, *Datalog and constraint satisfaction with infinite templates*, Journal on Computer and System Sciences **79** (2013), 79–100, A preliminary version appeared in the proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS'05).

[BFKR21] Manuel Bodirsky, Thomas Feller, Simon Knäuer, and Sebastian Rudolph, *On logics and homomorphism closure*, 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 2021, pp. 1–13.

[BG08] Manuel Bodirsky and Martin Grohe, *Non-dichotomies in constraint satisfaction complexity*, Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP) (Luca Aceto, Ivan Damgard, Leslie Ann

Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, eds.), Lecture Notes in Computer Science, Springer Verlag, July 2008, pp. 184 –196.

[Bir35] Garrett Birkhoff, *On the structure of abstract algebras*, Mathematical Proceedings of the Cambridge Philosophical Society **31** (1935), no. 4, 433–454.

[BJ03] Mathias Broxvall and Peter Jonsson, *Point algebras for temporal reasoning: Algorithms and complexity*, Artificial Intelligence **149** (2003), no. 2, 179–220.

[BJ17] Manuel Bodirsky and Peter Jonsson, *A model-theoretic view on qualitative constraint reasoning*, Journal of Artificial Intelligence Research **58** (2017), 339–385.

[BK02] Manuel Bodirsky and Martin Kutz, *Pure dominance constraints*, Proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS), 2002, pp. 287–298.

[BK07] ———, *Determining the consistency of partial tree descriptions*, Artificial Intelligence **171** (2007), 185–196.

[BK09a] Libor Barto and Marcin Kozik, *Constraint satisfaction problems of bounded width*, Proceedings of Symposium on Foundations of Computer Science (FOCS), 2009, pp. 595–603.

[BK09b] Manuel Bodirsky and Jan Kára, *The complexity of temporal constraint satisfaction problems*, Journal of the ACM **57** (2009), no. 2, 1–41, An extended abstract appeared in the Proceedings of the Symposium on Theory of Computing (STOC).

[BK12] Libor Barto and Marcin Kozik, *Absorbing subalgebras, cyclic terms and the constraint satisfaction problem*, Logical Methods in Computer Science **8/1** (2012), no. 07, 1–26.

[BK20] Manuel Bodirsky and Simon Knäuer, *Hardness of network satisfaction for relation algebras with normal representations*, Relational and Algebraic Methods in Computer Science, Springer International Publishing, 2020, pp. 31–46.

[BK21] Manuel Bodirsky and Simon Knäuer, *Network satisfaction for symmetric relation algebras with a flexible atom*, Proceedings of the AAAI Conference on Artificial Intelligence **35** (2021), no. 7, 6218–6226.

[BKJ05] Andrei A. Bulatov, Andrei A. Krokhin, and Peter G. Jeavons, *Classifying the complexity of constraints using finite algebras*, SIAM Journal on Computing **34** (2005), 720–742.

[BKKR69] V. G. Bodnarčuk, L. A. Kalužnin, V. N. Kotov, and B. A. Romov, *Galois theory for Post algebras, part I and II*, Cybernetics **5** (1969), 243–539.

[BKO$^+$19] Libor Barto, Michael Kompatscher, Miroslav Olšák, Trung Van Pham, and Michael Pinsker, *Equations in oligomorphic clones and the constraint satisfaction problem for ω-categorical structures*, Journal of Mathematical Logic **19** (2019), no. 2, #1950010, An extended abstract appeared at the Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science – LICS'17.

[BKR21] Manuel Bodirsky, Simon Knäuer, and Sebastian Rudolph, *Datalog - Expressibility for Monadic and Guarded Second-Order Logic*, 48th International Colloquium on Automata, Languages, and Programming (ICALP 2021) (Dagstuhl, Germany) (Nikhil Bansal, Emanuela Merelli, and James Worrell, eds.), Leibniz International Proceedings in Informatics (LIPIcs), vol. 198, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, pp. 120:1–120:17.

[BKS20] Manuel Bodirsky, Simon Knäuer, and Florian Starke, *Asnp: A tame fragment of existential second-order logic*, Beyond the Horizon of Computability, *Vol. 12098 of* Lecture Notes in Computer Science (Cham) (Marcella Anselmo, Gianluca Della Vedova, Florin Manea, and Arno Pauly, eds.), Springer International Publishing, 2020, pp. 149–162.

[BM11] Manuel Bodirsky and Jens K. Mueller, *Rooted phylogeny problems*, Logical Methods in Computer Science **7** (2011), no. 4, An extended abstract appeared in the proceedings of ICDT'10.

[BM16] Manuel Bodirsky and Antoine Mottet, *Reducts of finitely bounded homogeneous structures, and lifting tractability from finite-domain constraint satisfaction*, Proceedings of the 31th Annual IEEE Symposium on Logic in Computer Science (LICS), 2016, Preprint available at ArXiv:1601.04520, pp. 623–632.

[BMPP19] Manuel Bodirsky, Barnaby Martin, Michael Pinsker, and András Pongrácz, *Constraint satisfaction problems for reducts of homogeneous graphs*, SIAM Journal on Computing **48** (2019), no. 4, 1224–1264, A conference version appeared in the Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, pages 119:1–119:14.

[BN06] Manuel Bodirsky and Jaroslav Nešetřil, *Constraint satisfaction with countable homogeneous templates*, Journal of Logic and Computation **16** (2006), no. 3, 359–373.

[Bod08] Manuel Bodirsky, *Constraint satisfaction problems with infinite templates*, Complexity of Constraints (a collection of survey articles) (Heribert Vollmer, ed.), Lecture Notes in Computer Science, vol. 5250, Springer, 2008, pp. 196–228.

[Bod12] ______, *Complexity classification in infinite-domain constraint satisfaction*, Mémoire d'habilitation à diriger des recherches, Université Diderot – Paris 7. Available at arXiv:1201.0856v8, 2012.

[Bod15] ______, *Ramsey classes: Examples and constructions*, Surveys in Combinatorics. London Mathematical Society Lecture Note Series 424, Cambridge University Press, 2015, Invited survey article for the British Combinatorial Conference; ArXiv:1502.05146.

[Bod18] ______, *Finite relation algebras with normal representations*, Relational and Algebraic Methods in Computer Science - 17th International Conference, RAMiCS 2018, Groningen, The Netherlands, October 29 - November 1, 2018, Proceedings, 2018, pp. 3–17.

[Bod21] Manuel Bodirsky, *Complexity of infinite-domain constraint satisfaction*, Lecture Notes in Logic, Cambridge University Press, 2021.

[BOP18] Libor Barto, Jakub Opršal, and Michael Pinsker, *The wonderland of reflections*, Israel Journal of Mathematics **223** (2018), no. 1, 363–398.

[BP14] Manuel Bodirsky and Michael Pinsker, *Minimal functions on the random graph*, Israel Journal of Mathematics **200** (2014), no. 1, 251–296.

[BP15a] ______, *Schaefer's theorem for graphs*, Journal of the ACM **62** (2015), no. 3, 52 pages (article number 19), A conference version appeared in the Proceedings of STOC 2011, pages 655–664.

[BP15b] ______, *Topological Birkhoff*, Transactions of the American Mathematical Society **367** (2015), 2527–2549.

[BP20] Libor Barto and Michael Pinsker, *Topology is irrelevant*, SIAM Journal on Computing **49** (2020), no. 2, 365–393.

[BP21] Manuel Bodirsky and Michael Pinsker, *Canonical functions: a proof via topological dynamics*, Contributions Discret. Math. **16** (2021), no. 2, 36–45.

[BPP19] Manuel Bodirsky, Michael Pinsker, and András Pongrácz, *Projective clone homomorphisms*, The Journal of Symbolic Logic **86** (2019), no. 1, 148–161.

[BPR20] Manuel Bodirsky, Wied Pakusa, and Jakub Rydval, *Temporal constraint satisfaction problems in fixed-point logic*, Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (New York, NY, USA), LICS '20, Association for Computing Machinery, 2020, pp. 237–251.

[BPT11] Manuel Bodirsky, Michael Pinsker, and Todor Tsankov, *Decidability of definability*, Proceedings of the Symposium on Logic in Computer Science (LICS), 2011, pp. 321–328.

[BPT13] ———, *Decidability of definability*, Journal of Symbolic Logic **78** (2013), no. 4, 1036–1054, A conference version appeared in the Proceedings of LICS 2011.

[BR22] Manuel Bodirsky and Jakub Rydval, *On the descriptive complexity of temporal constraint satisfaction problems*, Journal of the ACM (2022).

[Bul03] Andrei A. Bulatov, *Tractable conservative constraint satisfaction problems*, Proceedings of the Symposium on Logic in Computer Science (LICS) (Ottawa, Canada), 2003, pp. 321–330.

[Bul11] Andrei A. Bulatov, *Complexity of conservative constraint satisfaction problems*, ACM Trans. Comput. Logic **12** (2011), no. 4.

[Bul16] Andrei A. Bulatov, *Conservative constraint satisfaction re-revisited*, Journal Computer and System Sciences **82** (2016), no. 2, 347–356, ArXiv:1408.3690.

[Bul17] ———, *A dichotomy theorem for nonuniform CSPs*, 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017, 2017, pp. 319–330.

[Cam99] Peter J. Cameron, *Permutation groups*, LMS Student Text 45, Cambridge University Press, Cambridge, 1999.

[CH04] Matteo Cristiani and Robin Hirsch, *The complexity of the constraint satisfaction problem for small relation algebras*, Artificial Intelligence Journal **156** (2004), 177–196.

[CL17] Hubie Chen and Benoît Larose, *Asking the metaquestions in constraint tractability*, TOCT **9** (2017), no. 3, 11:1–11:27.

[Com84] Stephen D. Comer, *Combinatorial aspects of relations*, Algebra Universalis **18** (1984), no. 1, 77–94.

[Con15] Gabriel J. Conant, *Model theory and combinatorics of homogeneous metric spaces*, Ph.D. thesis, University of Illinois at Chicago, 2015.

[Dün05] Ivo Düntsch, *Relation algebras and their application in temporal and spatial reasoning*, Artificial Intelligence Review **23** (2005), 315–357.

[Fra54] Roland Fraïssé, *Sur l'extension aux relations de quelques propriétés des ordres*, Annales Scientifiques de l'École Normale Supérieure **71** (1954), 363–388.

[Fra86] ______, *Theory of relations*, Elsevier Science Ltd, North-Holland, 1986.

[FV99] 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** (1999), 57–104.

[Gei68] David Geiger, *Closed systems of functions and predicates*, Pacific Journal of Mathematics **27** (1968), 95–100.

[GJ78] Michael Garey and David Johnson, *A guide to NP-completeness*, CSLI Press, Stanford, 1978.

[HG09] Paul Halmos and Steven Givant, *Introduction to boolean algebras*, Springer New York, 2009.

[HH01] R. Hirsch and I. Hodkinson, *Strongly representable atom structures of relation algebras*, Transactions of the American Mathematical Society **130** (2001), no. 6, 1819–1831).

[HH02] Robin Hirsch and Ian Hodkinson, *Relation algebras by games*, North Holland, 2002.

[Hir96] Robin Hirsch, *Relation algebras of intervals*, Artificial Intelligence Journal **83** (1996), 1–29.

[Hir97] ______, *Expressive power and complexity in algebraic logic*, Journal of Logic and Computation **7** (1997), no. 3, 309 – 351.

[Hir99] ______, *A finite relation algebra with undecidable network satisfaction problem*, Logic Journal of the IGPL **7** (1999), no. 4, 547–554.

[HJK19] Robin Hirsch, Marcel Jackson, and Tomasz Kowalski, *Algebraic foundations for qualitative calculi and networks*, Theor. Comput. Sci. **768** (2019), 99–116.

[HN19] Jan Hubička and Jaroslav Nešetřil, *All those Ramsey classes* (*Ramsey classes with closures and forbidden homomorphisms*), Advances in Mathematics **356** (2019), 106791.

[Hod97] Wilfrid Hodges, *A shorter model theory*, Cambridge University Press, Cambridge, 1997.

[JCG97] Peter Jeavons, David Cohen, and Marc Gyssens, *Closure properties of constraints*, Journal of the ACM **44** (1997), no. 4, 527–548.

[Jea98] Peter G. Jeavons, *On the algebraic structure of combinatorial problems*, Theoretical Computer Science **200** (1998), 185–204.

[Kar72] Richard M. Karp, *Reducibility among combinatorial problems*, Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations (Boston, MA), Springer US, 1972, pp. 85–103.

[Kaz15] Alexandr Kazda, *CSP for binary conservative relational structures*, Algebra Universalis **75** (2015), no. 1, 75–84.

[KKVW15] Marcin Kozik, Andrei Krokhin, Matt Valeriote, and Ross Willard, *Characterizations of several Maltsev conditions*, Algebra Universalis **73** (2015), no. 3, 205–224.

[Lad75] Richard E. Ladner, *On the structure of polynomial time reducibility*, Journal of the ACM **22** (1975), no. 1, 155–171.

[LM94] Peter B. Ladkin and Roger D. Maddux, *On binary constraint problems*, Journal of the Association for Computing Machinery **41** (1994), no. 3, 435–469.

[Lyn50] R. Lyndon, *The representation of relational algebras*, Annals of Mathematics **51** (1950), no. 3, 707–729.

[Mac11] Dugald Macpherson, *A survey of homogeneous structures*, Discrete Mathematics **311** (2011), no. 15, 1599–1634.

[Mad82] Roger Maddux, *Some varieties containing relation algebras*, Transactions of the American Mathematical Society **272** (1982), no. 2, 501–501.

[Mad94] Roger D. Maddux, *A perspective on the theory of relation algebras*, Algebra Universalis **31** (1994), no. 3, 456–465.

[Mad06a] ———, *Finite symmetric integral relation algebras with no 3-cycles*, Relations and Kleene Algebra in Computer Science, 9th International Conference on Relational Methods in Computer Science and 4th International Workshop on Applications of Kleene Algebra, RelMiCS/AKA 2006, Manchester, UK, August 29-September 2, 2006, Proceedings (Renate A. Schmidt, ed.), Lecture Notes in Computer Science, vol. 4136, Springer, 2006, pp. 2–29.

[Mad06b] Roger Duncan Maddux, *Relation algebras: Volume 150*, Studies in logic and the foundations of mathematics, Elsevier Science, London, England, May 2006 (en).

[MM08] Miklós Maróti and Ralph McKenzie, *Existence theorems for weakly symmetric operations*, Algebra Universalis **59** (2008), no. 3.

[MNPW21] Antoine Mottet, Tomáš Nagy, Michael Pinsker, and Michał Wrona, *Smooth Approximations and Relational Width Collapses*, 48th International Colloquium on Automata, Languages, and Programming (ICALP 2021) (Dagstuhl, Germany) (Nikhil Bansal, Emanuela Merelli, and James Worrell, eds.), Leibniz International Proceedings in Informatics (LIPIcs), vol. 198, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, pp. 138:1–138:20.

[MP22] Antoine Mottet and Michael Pinsker, *Smooth approximations and csps over finitely bounded homogeneous structures*, Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science (New York, NY, USA), LICS '22, Association for Computing Machinery, 2022.

[NR89] Jaroslav Nešetřil and Vojtěch Rödl, *The partite construction and Ramsey set systems*, Discrete Mathematics **75** (1989), no. 1-3, 327–334.

[PK79] Reinhard Pöschel and Lev A. Kalužnin, *Funktionen- und relationenalgebren*, Deutscher Verlag der Wissenschaften, 1979.

[Pos41] Emil L. Post, *The two-valued iterative systems of mathematical logic*, Annals of Mathematics Studies **5** (1941).

[PR96] Maurice Pouzet and Bernard Roux, *Ubiquity in category for metric spaces and transition systems*, European Journal of Combinatorics **17** (1996), no. 2, 291–307.

[RN99] Jochen Renz and Bernhard Nebel, *On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus*, Artificial Intelligence **108** (1999), no. 1-2, 69–123.

[RN07] ———, *Qualitative spatial reasoning using constraint calculi*, Handbook of Spatial Logics (M. Aiello, I. Pratt-Hartmann, and J. van Benthem, eds.), Springer Verlag, Berlin, 2007, pp. 161–215.

[Sch78] Thomas J. Schaefer, *The complexity of satisfiability problems*, Proceedings of the Symposium on Theory of Computing (STOC), 1978, pp. 216–226.

[Sig10] Mark H. Siggers, *A strong Mal'cev condition for varieties omitting the unary type*, Algebra Universalis **64** (2010), no. 1, 15–20.

[Tar41] Alfred Tarski, *On the calculus of relations*, Journal of Symbolic Logic **6** (1941), no. 3, 73–89.

[Tay77] Walter Taylor, *Varieties obeying homotopy laws*, Canadian Journal of Mathematics **29** (1977), 498–527.

[VKv90] Marc Vilain, Henry Kautz, and Peter van Beek, *Constraint propagation algorithms for temporal reasoning: A revised report*, Readings in Qualitative Reasoning

About Physical Systems (Daniel S. Weld and Johan de Kleer, eds.), Morgan Kaufmann, 1990, pp. 373–381.

[Wro20a] Michał Wrona, *On the relational width of first-order expansions of finitely bounded homogeneous binary cores with bounded strict width*, Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (New York, NY, USA), LICS '20, Association for Computing Machinery, 2020, pp. 958–971.

[Wro20b] ———, *Relational Width of First-Order Expansions of Homogeneous Graphs with Bounded Strict Width*, 37th International Symposium on Theoretical Aspects of Computer Science (STACS 2020) (Dagstuhl, Germany) (Christophe Paul and Markus Bläser, eds.), Leibniz International Proceedings in Informatics (LIPIcs), vol. 154, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020, pp. 39:1–39:16.

[Zhu17] Dmitriy N. Zhuk, *A proof of CSP dichotomy conjecture*, 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017, 2017, pp. 331–342.

[Zhu20] ———, *A proof of the csp dichotomy conjecture*, J. ACM **67** (2020), no. 5.

List of Figures

Versicherung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Die vorliegende Dissertation wurde in der Zeit von Oktober 2018 bis Oktober 2022 am Institut für Algebra der TU Dresden unter der wissenschaftlichen Betreuung von Prof. Dr. Manuel Bodirsky angefertigt.

Simon Knäuer
Dresden, den 20. Oktober 2022.