2013

Commutative, idempotent groupoids and the constraint satisfaction problem

David Michael Failing
Iowa State University

Commutative, idempotent groupoids and the constraint satisfaction problem

by

David Michael Failing

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Mathematics

Program of Study Committee:

Clifford Bergman, Major Professor

Elgin Johnston

Roger Maddux

Jonathan D.H. Smith

Sung-Yell Song

Iowa State University

Ames, Iowa

2013

DEDICATION

I would like to dedicate this thesis to my parents. Mom—your patience and desire to understand just what it means to complete a Ph.D. have helped keep me focused on the bigger picture. Dad—your wisdom from years working in business has proven invaluable and transferrable to just about any struggle I've come up against in graduate school. Without the love and support received from both of you, I could not have made it this far.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

First, I would like to thank Professor Cliff Bergman for his patience and guidance over the years. Whatever the task I was focused on, Dr. Bergman was right alongside, working just as hard as I was. Without your dedication to my success, I could not have arrived at this point.

I would also like to acknowledge the assistance of several others outside Iowa State. Petr Vojtěchovský suggested the Bol-Moufang identities as a direction of study, and they turned out to be a rich source of structural properties. An REU project supervised by his coauthor, J.D. Phillips, inspired some further generalizations in my work. Lastly, Peter Jipsen's talk at the 2011 AMS Central Section Meeting in Iowa City inspired me to approach the classification component of my thesis work using automated reasoning tools. I was unsure of their utility at first, but Peter's assistance helped me to approach some of my work from an empirical angle, examining data for patterns, something I had not previously thought applicable in universal algebra.

Finally, I would like to thank my friends and colleagues, who listened to my musings, complaints, and thinking out loud as I grappled with my thesis problem. Your patience and encouragement helped me through the day-to-day struggles of graduate school. Thank you all for your contributions to my success.

CHAPTER 1. BACKGROUND MATERIAL

1.1 Introduction

The goal in a Constraint Satisfaction Problem (CSP) is to determine if there is a suitable assignment of values to variables subject to constraints on their allowed simultaneous values. The CSP provides a common framework in which many important combinatorial problems may be formulated—for example, graph colorability or propositional satisfiability. It is also of great importance in theoretical computer science, where it is applied to problems as varied as database theory and natural language processing.

In what follows, we will assume $\mathbf{P} \neq \mathbf{NP}$. Problems in **P** are said to be tractable. The general CSP is known to be **NP**-complete [30]. One focus of current research is on instances of the CSP in which the constraint relations are members of some fixed finite set of relations over a finite set. The goal is then to characterize the computational complexity of the CSP based upon properties of that set of relations. Feder and Vardi [14] studied broad families of constraints which lead to a tractable CSP. Their work inspired what is known as the CSP Dichotomy Conjecture, postulating that every fixed set of constraint relations is either **NP**-complete or tractable.

A discovery of Jeavons, Cohen, and Gyssens [20], later refined by Bulatov, Jeavons and Krokhin [9] was the ability to translate the question of the complexity of the CSP over a set of relations to a question of algebra. Specifically, they showed that the complexity of any particular CSP depends solely on the *polymorphisms* of the constraint relations, that is, the functions preserving all the constraints. The translation to universal algebra was made complete by Bulatov, Jeavons, and Krokhin in recognizing that to each CSP, one can associate an algebra whose operations consist of the polymorphisms of the constraints. Following this, the

Dichotomy Conjecture of Feder and Vardi was recast as the *Algebraic Dichotomy Conjecture*, a condition with a number of equivalent statements (summarized in [10]) which suggests a sharp dividing line between those CSPs that are **NP**-complete and those that are tractable, dependent solely upon universal algebraic conditions of the associated algebra. One of these conditions is the existence of a weak near-unanimity term (WNU, see Definition 1.3.3). Roughly speaking, the Algebraic Dichotomy Conjecture asserts that an algebra corresponds to a tractable CSP if and only if it has a WNU term. The necessity of this condition was established in [9]. Our goal in this thesis is to provide further evidence of sufficiency.

It follows easily from Definition 1.3.3 that a binary operation is weak near-unanimity if and only if it is commutative and idempotent. This motivates us to consider algebras with a single binary operation that is commutative and idempotent—CI-groupoids for short. If the Algebraic Dichotomy Conjecture is true, then every finite CI-groupoid will give rise to a tractable CSP.

In [20] it was proved that the dichotomy conjecture holds for CI-groupoids that are associative, in other words, for semilattices. This result was generalized in [6] by weakening associativity to the identity $x(xy) \approx xy$. In the present work we continue this line of attack by considering several other identities that (in the presence of commutativity and idempotence) are strictly weaker than associativity.

The earliest sections of the thesis are devoted to supporting material. In the present chapter, we review the necessary concepts of universal algebra and constraint satisfaction. In Chapter 2, we discuss the Płonka sum, as well as a generalization which we will use as our primary structural tool. The generalization is applied to obtain a general preservation result for tractable CSPs. We are hopeful that this technique will prove useful in future analysis of constraint satisfaction. A family of identities weaker than the associative law, those of Bol-Moufang type, is studied in Chapter 3. In Chapter 4, we analyze CI-groupoids satisfying the self-distributive law $x(yz) \approx (xy)(xz)$, entropic law, and other generalizations of associativity. In addition to proving that each of these conditions implies tractability, we establish some structure theorems that may be of further interest. The tractability results in this paper are related to some unpublished work of Maróti [33, 34]. On the whole, our results and his seem to be incomparable. The final chapter outlines some possible directions for future research, and two appendices discuss (and present) equational derivations obtained with automated reasoning tools.

1.2 Universal Algebra

An *algebra* $\mathbf{A}$ is a pair $\langle A, \mathcal{F}^{\mathbf{A}} \rangle$, where A is a nonempty set (the *universe* of $\mathbf{A}$), $\mathcal{F}$ is a family of *operation symbols*, and $\mathcal{F}^{\mathbf{A}} = \langle f^{\mathbf{A}} \colon f \in \mathcal{F} \rangle$ is a family of operations on A, known as the *basic operations* of the algebra. The algebra $\mathbf{A} = \langle A, \mathcal{F}^{\mathbf{A}} \rangle$ has a corresponding function $\rho \colon \mathcal{F} \to \mathbb{N}$ which assigns to each $f \in \mathcal{F}$ the *rank* or *arity* of $f^{\mathbf{A}}$. This function is known as the *similarity type* of $\mathbf{A}$, or simply the *type*. Algebras of the same type are said to be *similar*. Operations of rank 0, 1 or 2 are called *nullary* or *constant*, *unary*, and *binary*, respectively. We will leave off superscripts from the operations unless they are needed for clarity. An algebra whose universe consists of a single element is said to be *trivial*. We begin with some examples of algebras which will be used throughout this work.

Definition 1.2.1. A *groupoid* is an algebra $\langle G, \cdot \rangle$ with a single binary operation. Sometimes such an algebra is referred to as a *binar* or *magma*.

Definition 1.2.2. A *Latin square* is a groupoid $\langle G, \cdot \rangle$ such that the equation $x \cdot y \approx z$ has a unique solution whenever two of the three variables are specified.

Sometimes we wish to define one algebra based upon another. If $\mathbf{B} = \langle A, \mathcal{G} \rangle$ and $\mathbf{A} = \langle A, \mathcal{F} \rangle$ are two algebras such that $\mathcal{G}$ is a subsequence of $\mathcal{F}$, we refer to $\mathbf{B}$ as a *reduct* of $\mathbf{A}$, and call $\mathbf{A}$ an *expansion* of $\mathbf{B}$.

Definition 1.2.3. A *quasigroup* is an algebra $\langle A, \cdot, /, \backslash \rangle$ with three binary operations satisfying the identities

$$\begin{aligned} x\backslash(x \cdot y) &\approx y, \qquad (x \cdot y)/y \approx x, \\ x \cdot (x\backslash y) &\approx y, \qquad (x/y) \cdot y \approx x. \end{aligned} \tag{1.1}$$

Quasigroups are a (not necessarily associative) generalization of groups.

If $\langle A, \cdot, /, \backslash \rangle$ is a quasigroup, then its reduct $\langle A, \cdot \rangle$ is a Latin square. Conversely, every Latin square $\langle A, \cdot \rangle$ has an expansion to a quasigroup by defining $a\backslash b$ and b/a to be the unique solutions x, y of $a \cdot y = b$ and $x \cdot a = b$, respectively. A quasigroup with an *identity element* e such that $x \cdot e \approx e \cdot x \approx x$ is known as a *loop*.

Definition 1.2.4. A *semilattice* is a groupoid $\langle S, \vee \rangle$ satisfying the associative law

$$x \vee (y \vee z) \approx (x \vee y) \vee z,$$

the commutative law

$$x \vee y \approx y \vee x,$$

and the idempotent law

$$x \vee x \approx x.$$

A *lattice* is an algebra $\langle L, \wedge, \vee \rangle$ with two binary operations such that both $\langle L, \wedge \rangle$ and $\langle L, \vee \rangle$ are semilattices, and such that the basic operations satisfy the absorption laws

$$x \wedge (x \vee y) \approx x \qquad x \vee (x \wedge y) \approx x.$$

Definition 1.2.5. A *group* is an algebra $\langle G, \cdot, ^{-1}, e \rangle$ of type $\langle 2, 1, 0 \rangle$ such that $\langle G, \cdot \rangle$ is an associative groupoid, and satisfying $x \cdot x^{-1} \approx x^{-1} \cdot x \approx e$ and $x \cdot e \approx e \cdot x \approx x$. $\mathbf{G}$ is an *Abelian* or *commutative* group if the operation $\cdot$ is commutative.

A *ring* is an algebra $\langle R, +, \cdot, -, 0 \rangle$ such that $\langle R, +, -, 0 \rangle$ is an Abelian group, $\langle R, \cdot \rangle$ is an associative groupoid satisfying the distributive laws

$$x \cdot (y + z) \approx x \cdot y + x \cdot z$$

$$(y + z) \cdot x \approx y \cdot x + z \cdot x.$$

Definition 1.2.6. For a fixed ring $\mathbf{R}$, an $\mathbf{R}$-*module* is an algebra $\langle M, +, -, 0, \langle r \colon r \in R \rangle \rangle$ where each $r \in R$ is interpreted as a unary operation, $\langle M, +, -, 0 \rangle$ is an Abelian group and satisfying, for each $r, s \in R$, the equations

$$r(x + y) \approx rx + ry$$

$$(r + s)x \approx rx + sx$$

$$r(sx) \approx (rs)x$$

Definition 1.2.7. Let $\mathbf{A} = \langle A, \mathcal{F}^{\mathbf{A}} \rangle$, $\mathbf{B} = \langle B, \mathcal{F}^{\mathbf{B}} \rangle$ and $\{\mathbf{A}_i = \langle A_i, \mathcal{F}^{\mathbf{A}_i} \rangle \mid i \in I\}$ be algebras of type $\rho\colon \mathcal{F} \to \mathbb{N}$.

1. A function $h\colon B \to A$ is a *homomorphism* from $\mathbf{B}$ to $\mathbf{A}$ if it respects all of the operations of the algebras, i.e. for $f \in \mathcal{F}$ a basic n-ary operation, and $b_1, \ldots, b_n \in B$,

$$h(f^{\mathbf{B}}(b_1, \ldots, b_n)) = f^{\mathbf{A}}(h(b_1), \ldots, h(b_n)).$$

   We say that $\mathbf{A}$ is a *homomorphic image* of $\mathbf{B}$ if there is a homomorphism from $\mathbf{B}$ to $\mathbf{A}$ which is onto.

2. $\mathbf{B}$ is a *subalgebra* of $\mathbf{A}$ (denoted $\mathbf{B} \leq \mathbf{A}$) if $B \subseteq A$ and for every $f \in \mathcal{F}$, $f^{\mathbf{B}} = f^{\mathbf{A}}\big|_{B^{\rho(f)}}$. That is, if B is closed under all the operations of $\mathbf{A}$.

3. The *direct product* of the algebras $\langle \mathbf{A}_i \mid i \in I \rangle$ is the algebra $\prod_{i \in I} \mathbf{A}_i$ with universe $\prod_{i \in I} A_i$ and basic operations defined component-wise.

4. We say that $\mathbf{A}$ is a *subdirect product* of the algebras $\langle \mathbf{A}_i \mid i \in I \rangle$ if $\mathbf{A} \leq \prod_{i \in I} \mathbf{A}_i$ and for every $i \in I$, the projection $\pi_i\colon \mathbf{A} \to \mathbf{A}_i$ is surjective. A nontrivial algebra $\mathbf{A}$ is *subdirectly irreducible* if whenever $\mathbf{A}$ can be written as a subdirect product of the algebras $\langle \mathbf{A}_i \mid i \in I \rangle$, some π_i is an isomorphism.

Each of the notions above corresponds to a particular closure operator. Given a class $\mathcal{K}$ of similar algebras, we adopt the notation

$\mathsf{H}(\mathcal{K}) =$ the class of all homomorphic images of members of $\mathcal{K}$.

$\mathsf{S}(\mathcal{K}) =$ the class of all algebras isomorphic to subalgebras of members of $\mathcal{K}$.

$\mathsf{P}(\mathcal{K}) =$ the class of all algebras isomorphic to direct products of members of $\mathcal{K}$.

$\mathsf{S_P}(\mathcal{K}) =$ the class of all algebras isomorphic to subdirect products of members of $\mathcal{K}$.

A *variety* $\mathcal{V}$ is a class of similar algebras which is closed under each of H, S and P. The smallest variety of any given similarity type is the set containing a trivial algebra of that type. For $\mathcal{K}$ a class of similar algebras, we define the additional closure operator $\mathsf{V}(\mathcal{K})$ to be the smallest variety containing $\mathcal{K}$. $\mathsf{V}(\mathcal{K})$ is called the *variety generated by* $\mathcal{K}$. Garrett Birkhoff's

HSP Theorem showed that to generate the variety $\mathsf{V}(\mathcal{K})$, it is enough to apply each of **H**, **S** and **P** to $\mathcal{K}$ once, provided it is done in the correct order.

Theorem 1.2.8 ([4]). $\mathsf{V} = \mathsf{HSP}$.

Later, Birkhoff showed in the Subdirect Representation Theorem [5] that every algebra is isomorphic to a subdirect product of subdirectly irreducible algebras. For $\mathcal{K}$ a class of similar algebras, let $\mathcal{K}_{\text{si}}$ denote the collection of subdirectly irreducible members of $\mathcal{K}$. The Subdirect Representation Theorem implies the following result.

Theorem 1.2.9 ([5]). *Let $\mathcal{V}$ be a variety. Then $\mathcal{V} = \mathsf{S}_{\mathsf{P}}(\mathcal{V}_{si})$. That is, every variety $\mathcal{V}$ is the class of all algebras isomorphic to subdirect products of subdirectly irreducible members of $\mathcal{V}$.*

Another way to view homomorphic images is through the lens of *congruence relations.* Given a set A, an n-ary relation over A is simply a subset $R \subseteq A^n$ of the n-th power of A. Relations are central both to universal algebra and its connection to the CSP.

Definition 1.2.10. Let $\mathbf{A}$ be an algebra, and R a binary relation on A. We say that R has the *substitution property* if for every basic n-ary operation f of $\mathbf{A}$,

$$(a_1, b_1), \dots, (a_n, b_n) \in R \Rightarrow (f(a_1, \dots, a_n), f(b_1, \dots, b_n)) \in R.$$

We say that a binary relation R is a *congruence relation on A* if it is an equivalence relation with the substitution property.

Typically, congruences are represented by Greek letters such as θ or ψ, and more general relations are given as R, S, T, etc. Now, given any homomorphism $h\colon \mathbf{A} \to \mathbf{B}$, the relation

$$\ker f = \{(a_1, a_2) \in A^2 \mid h(a_1) = h(a_2)\}$$

is a congruence on $\mathbf{A}$. Conversely, given a congruence relation θ on $\mathbf{A}$, the map

$$q_\theta\colon \mathbf{A} \to \mathbf{A}/\theta;\ a \mapsto a/\theta$$

gives the quotient $\mathbf{A}/\theta$ as a homomorphic image of $\mathbf{A}$. The basic operations on $\mathbf{A}/\theta$ are defined implicitly by the homomorphism condition as

$$f^{\mathbf{A}/\theta}(a_1/\theta, \dots, a_n/\theta) = f^{\mathbf{A}}(a_1, \dots, a_n)/\theta.$$

The collection of all congruences on an algebra $\mathbf{A}$ is a lattice (denoted $\operatorname{Con}\mathbf{A}$) with operations $\theta \wedge \psi = \theta \cap \psi$ and $\theta \vee \psi$ the smallest congruence containing both θ and ψ.

1.3 Terms and Equations

While one focus of universal algebra is the study of *structural* properties of algebras and varieties, using the basic notions of the previous section, another focus which will be of great importance in the present work is the study of *semantic* properties. In particular, the link between the semantic and the structural will be used to shed light on the CSP.

Definition 1.3.1. Let $\rho\colon \mathcal{F} \to \mathbb{N}$ be a similarity type, and X a countable set of *variables*, disjoint from $\mathcal{F}$. We define the *terms of type* ρ recursively as follows:

1. Every variable is a term.
2. If $f \in \mathcal{F}$ with $\rho(f) = n$ and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term.

For any algebra $\mathbf{A}$ and term $t(x_1, \ldots, x_n)$ of the same type, we define the an n-ary operation $t^{\mathbf{A}}(x_1, \ldots, x_n)$, the *term operation on* $\mathbf{A}$ *induced by* t recursively as follows:

1. If t is the variable x_i, then $t^{\mathbf{A}}(x_1, \ldots, x_n) = x_i$.
2. If $t(x_1, \ldots, x_n) = f(t_1(x_1, \ldots, x_n), \ldots, t_m(x_1, \ldots, x_n))$, then

$$t^{\mathbf{A}}(x_1, \ldots, x_n) = f^{\mathbf{A}}(t_1^{\mathbf{A}}(x_1, \ldots, x_n), \ldots, t_m^{\mathbf{A}}(x_1, \ldots, x_n)).$$

The set of all term operations on an algebra $\mathbf{A}$ will be denoted $\operatorname{Term}(\mathbf{A})$. An *equation* or *identity* is an expression of the form $p(x_1, \ldots, x_n) \approx q(x_1, \ldots, x_n)$ (often shortened to $p \approx q$), where $p(x_1, \ldots, x_n)$ and $q(x_1, \ldots, x_n)$ are terms of the same type. We say that an algebra $\mathbf{A}$ *satisfies* identity $p \approx q$ if the term operations $p^{\mathbf{A}}$ and $q^{\mathbf{A}}$ are equal as functions; a class $\mathcal{K}$ of algebras satisfies an identity $p \approx q$ if every algebra in the class satisfies $p \approx q$. For a set Σ of equations and a class $\mathcal{K}$ of algebras, we say that $\mathcal{K}$ satisfies Σ if $\mathcal{K}$ satisfies every identity in Σ. We define $\operatorname{Mod}(\Sigma)$ to be the class of algebras satisfying Σ, and $\operatorname{Id}(\mathcal{K})$ to be the class of identities satisfied by $\mathcal{K}$.

The examples of algebras provided in the previous section were given as structures whose basic operations satisfied certain axioms. In 1935, Birkhoff [4] showed that a class of algebras is a variety if and only if it is $\mathrm{Mod}\,(\Sigma)$ for some set Σ of identities (called an *equational base* for the variety). Since they were defined in the previous section as algebras satisfying certain equations, the classes of all groupoids, quasigroups, groups, etc. each form varieties. The connection between structural results and semantic results in universal algebra comes via *Maltsev conditions*. That is, the existence of special term operations that satisfy a particular set of equations. Several such conditions will be used in the remainder of this work.

Definition 1.3.2. For $k \geq 2$, a *k-edge operation* on a set A is a $(k+1)$-ary operation, f, on A satisfying the k identities:

$$\begin{aligned} f(x,x,y,y,y,\ldots,y,y) &\approx y \\ f(x,y,x,y,y,\ldots,y,y) &\approx y \\ f(y,y,y,x,y,\ldots,y,y) &\approx y \\ f(y,y,y,y,x,\ldots,y,y) &\approx y \\ &\vdots \\ f(y,y,y,y,y,\ldots,x,y) &\approx y \\ f(y,y,y,y,y,\ldots,y,x) &\approx y \end{aligned}$$

Definition 1.3.3. An operation f is *idempotent* if it satisfies $f(x,\ldots,x) \approx x$. A k-ary *weak near-unanimity operation* on A is an idempotent operation that satisfies the identities

$$f(y,x,\ldots,x) \approx f(x,y,\ldots,x) \approx \cdots \approx f(x,x,\ldots,x,y).$$

A k-ary *near-unanimity operation* is a weak near-unanimity operation that satisfies the identity $f(y,x,\ldots,x) \approx x$.

Definition 1.3.4. A *Maltsev operation* on a set A is a ternary operation $q(x,y,z)$ satisfying $q(x,y,y) \approx q(y,y,x) \approx x$.

Some term conditions for an algebra, such as the existence of a Maltsev term, are equivalent to certain properties of the congruence lattice of the algebra. Given two relations θ and ψ, we define a new relation $\theta \circ \psi$ (the *relative product* of θ and ψ) by

$$\theta \circ \psi = \{(x, z) \mid (\exists y)\,(x, y) \in \theta \text{ and } (y, z) \in \psi\}.$$

The congruences θ and ψ are said to *permute* if $\theta \circ \psi = \psi \circ \theta$. If θ and ψ are permuting congruences on an algebra $\mathbf{A}$, then in Con $\mathbf{A}$, $\theta \vee \psi = \theta \circ \psi$. An algebra is said to be *congruence-permutable* if every pair of congruences on the algebra permutes, while a variety is said to be congruence-permutable if every one of its members is congruence-permutable. The following theorem, due to Maltsev, provides our first example of the link between the structural and semantic sides of the subject.

Theorem 1.3.5 ([31]). *Let $\mathcal{V}$ be a variety of algebras. The following are equivalent:*

(a) $\mathcal{V}$ is congruence-permutable.

(b) $\mathcal{V}$ has a Maltsev term. That is, a ternary term q such that $\mathcal{V}$ satisfies

$$q(x, y, y) \approx q(y, y, x) \approx x.$$

Many other results exist which link lattice-theoretic properties of the congruence lattices of algebras (distributivity, modularity, etc.) to the satisfaction of certain Maltsev conditions, and are discussed in Section 4.7 of [3].

An algebra is said to be *congruence meet-semidistributive* (SD($\wedge$)) if its congruence lattice satisfies the implication

$$(x \wedge y \approx x \wedge z) \Rightarrow (x \wedge (y \vee z) \approx x \wedge y).$$

A variety $\mathcal{V}$ is congruence meet-semidistributive if every algebra in $\mathcal{V}$ is congruence meet-semidistributive. A Maltsev condition is said to be *idempotent* if every term involved in the condition is idempotent. Kearnes and Kiss provide an extensive theorem, with nine conditions each equivalent to congruence meet-semidistributivity. We will need just one of them for our purposes.

Theorem 1.3.6 ([23, Theorem 8.1]). *Let $\mathcal{V}$ be a variety of algebras. The following are equivalent:*

(a) $\mathcal{V}$ *is congruence meet-semidistributive.*

(b) $\mathcal{V}$ *satisfies a family of idempotent Maltsev conditions that, considered together, fail in any nontrivial variety of modules.*

1.4 CSP Definitions and Theorems

In order to achieve our main result, we must collect together several notions of the CSP (largely outlined in [8]), and ways of moving between them. We also survey the major algorithms at our disposal to establish the tractability of particular classes of CSPs.

Definition 1.4.1. An *instance* of the CSP is a triple $\mathcal{R} = (V, A, \mathcal{C})$ in which:

- V is a finite set of *variables*,
- A is a nonempty, finite set of *values*,
- $\mathcal{C} = \{(S_i, R_i) \mid i = 1, \dots, n\}$ is a set of *constraints*, with each S_i an m_i-tuple of variables, and each R_i an m_i-ary relation over A which indicates the allowed simultaneous values for variables in S_i.

Given an instance $\mathcal{R}$ of the CSP, we wish to answer the question: Does $\mathcal{R}$ have a *solution*? That is, is there a map $f\colon V \to A$ such that for $1 \leq i \leq n$, $f(S_i) \in R_i$?

The class of all CSP instances is **NP**-complete but, by restricting the form of the constraint relations, we can identify certain subclasses which are tractable.

Definition 1.4.2. Let Γ be a set of finitary relations over a set A. CSP(Γ) denotes the decision problem whose instances have set of values A and constraint relations coming from Γ.

We refer to this first notion of the CSP as *single-sorted*. A common example of the single-sorted CSP(Γ) is the graph k-colorability problem, given by $\Gamma = \{\neq_A\}$, where $\neq_A$ is the binary disequality relation on any set with $|A| = k$. An assortment of other examples are presented in [9] and [20].

A second formulation of the CSP arises naturally in the context of conjunctive queries to relational databases (for more information about the connection see [8, Definition 2.7]). For a class of sets $\mathcal{A} = \{A_i \mid i \in I\}$, a subset R of $A_{i_1} \times \cdots \times A_{i_k}$ together with the list of indices $(i_1, \ldots, i_k)$ is called a *k-ary relation over $\mathcal{A}$ with signature* $(i_1, \ldots, i_k)$.

Definition 1.4.3. An *instance* of the *many-sorted* CSP is a quadruple $\mathcal{R} = (V, \mathcal{A}, \delta, \mathcal{C})$ in which:

- V is a finite set of *variables*,
- $\mathcal{A} = \{A_i \mid i \in I\}$ is a collection of finite sets of *values*,
- $\delta \colon V \to I$ is called the *domain function*,
- $\mathcal{C} = \{(S_i, R_i) \mid i = 1, \ldots, n\}$ is a set of *constraints*. For $1 \leq i \leq n$, $S_i = (v_1, \ldots, v_{m_i})$ is an m_i-tuple of variables, and each R_i is an m_i-ary relation over $\mathcal{A}$ with signature $(\delta(v_1), \ldots, \delta(v_{m_i}))$ which indicates the allowed simultaneous values for variables in S_i.

Given an instance $\mathcal{R}$ of the many-sorted CSP, we wish to answer the question: Does $\mathcal{R}$ have a *solution*? That is, is there a map $f \colon V \to \bigcup_{i \in I} A_i$ such that for each $v \in V$, $f(v) \in A_{\delta(v)}$, and for $1 \leq i \leq n$, $f(S_i) \in R_i$?

The single-sorted version of the CSP is obtained from the many-sorted by requiring the domain function δ to be constant. It is tacitly assumed that every instance of a constraint satisfaction problem can be encoded as a finite binary string. The length of that string is formally considered to be the size of the instance. We can restrict our attention to specific classes of the many-sorted CSP in a manner similar to the one we used in the single-sorted case.

Definition 1.4.4. Let Γ be a set of relations over the class of sets $\mathcal{A} = \{A_i \mid i \in I\}$. CSP($\Gamma$) denotes the decision problem with instances of the form $(V, \mathcal{B}, \delta, \mathcal{C})$ in which $\mathcal{B} \subseteq \mathcal{A}$ and every constraint relation is a member of Γ.

In either case (many- or single-sorted), we are concerned with determining which sets of relations result in a tractable decision problem.

Definition 1.4.5. Let Γ be a set of relations. We say that Γ is *tractable* if for every finite subset $\Delta \subseteq \Gamma$, the class CSP(Δ) lies in **P**. If there is some finite $\Delta \subseteq \Gamma$ for which CSP(Δ) is **NP**-complete, we say that Γ is ***NP**-complete*.

The above notion of tractability is referred to in the literature as *local tractability*, since the tractability of each particular finite subset Δ may depend on a distinct polynomial-time algorithm for its CSP. The idea of *global tractability*, that a single algorithm exists which solves CSP(Δ) for every finite subset Δ, is clearly at least as strong as local tractability. It is postulated that the two notions are equivalent.

Building upon Schaefer's earlier dichotomy result for CSPs over a two-element domain [45], Feder and Vardi [14] conjectured that every finite set of relations is either tractable or **NP**-complete. Jeavons and his coauthors [8, 9, 19, 20] later made explicit the link between families of relations over finite sets and finite algebras that has made possible many partial solutions to this so-called Dichotomy Conjecture. In order to complete the transition from sets of relations to finite algebras, we collect a few more definitions.

Definition 1.4.6. Let A be a set, Γ a set of finitary relations on A, F a set of finitary operations on A, R an n-ary relation on A, and f an m-ary operation on A.

1. We say that f is a *polymorphism* of R, or that R is *invariant* under f (see [3, Definition 4.11]) if
$$\overline{a}_1, \ldots, \overline{a}_m \in R \Rightarrow f(\overline{a}_1, \ldots, \overline{a}_m) \in R.$$
2. $\mathrm{Pol}(\Gamma) = \{f \mid f \text{ preserves every } R \in \Gamma\}$, the *clone of polymorphisms* of Γ.
3. $\mathrm{Inv}(\mathcal{F}) = \{R \mid R \text{ is invariant under every } f \in \mathcal{F}\}$, the *relations invariant under* $\mathcal{F}$.
4. $\langle\Gamma\rangle$ denotes $\mathrm{Inv}(\mathrm{Pol}(\Gamma))$, the *relational clone on* A *generated by* Γ.

The following result ([9, Corollary 2.17]) relates the computational complexity of a set of finitary relations to the complexity of the relational clone it generates.

Theorem 1.4.7. *Let* Γ *be a set of finitary relations on the finite set* A. Γ *is tractable if and only if* $\langle\Gamma\rangle$ *is tractable. Similarly,* Γ *is* **NP***-complete if and only of* $\langle\Gamma\rangle$ *is* **NP***-complete.*

To every set of relations Γ over a finite set A, we can associate the finite algebra $\mathbf{A}_\Gamma = \langle A, \mathrm{Pol}(\Gamma)\rangle$. Likewise, to every finite algebra $\mathbf{A} = \langle A, \mathcal{F}\rangle$, we can associate the set of relations $\mathrm{Inv}(\mathcal{F})$. We call an algebra $\mathbf{A} = \langle A, \mathcal{F}\rangle$ tractable (**NP**-complete) precisely when $\mathrm{Inv}(\mathcal{F})$ is a tractable (**NP**-complete) set of relations, and write CSP($\mathbf{A}$) to denote the decision problem CSP(Inv($\mathcal{F}$)). In fact, combining Theorem 1.4.7 with the fact that $\langle\Gamma\rangle = \text{Inv}(\text{Pol}(\Gamma))$ suggests that the Algebraic Dichotomy Conjecturemight be settled by restricting one's attention to algebras.

Ultimately, it is enough to restrict our attention to *idempotent* algebras. That is, those algebras whose basic operations are all idempotent. To see this, we first define an algebra to be *surjective* if all of its term operations are surjective. Bulatov, Jeavons and Krokhin begin the restriction to the idempotent case as follows.

Definition 1.4.8. Let $\mathbf{A} = \langle A, \mathcal{F}\rangle$ be an algebra, and let U be a nonempty subset of A. The *term induced algebra* $\mathbf{A}|_U$ is defined as $\langle U, \text{Term}(\mathbf{A})|_U\rangle$, where

$$\text{Term}(\mathbf{A})|_U = \{g|_U \colon g \in \text{Term}(\mathbf{A}) \text{ and } U \text{ is invariant under } g\}.$$

Theorem 1.4.9 ([9, Theorem 4.4])**.** *Let* $\mathbf{A} = \langle A, \mathcal{F}\rangle$ *be a finite algebra. There is a subset* $U \subseteq A$ *such that* $\mathbf{A}|_U$ *is surjective, and* $\mathbf{A}$ *is tractable (*$\mathbf{NP}$*-complete) if and only if* $\mathbf{A}|_U$ *is tractable (*$\mathbf{NP}$*-complete).*

In order to settle the dichotomy using universal algebra, then, it is enough to to look at surjective algebras. However, it suffices to consider only special surjective algebras, namely those which are also idempotent.

Definition 1.4.10. The *full idempotent reduct* of an algebra $\mathbf{A} = \langle A, \mathcal{F}\rangle$ is the algebra $\langle A, \text{Term}_{\text{id}}(\mathbf{A})\rangle$, where $\text{Term}_{\text{id}}(\mathbf{A})$ consists of all the idempotent term operations on $\mathbf{A}$.

Theorem 1.4.11 ([9, Theorem 4.7])**.** *A finite surjective algebra* $\mathbf{A}$ *is tractable (*$\mathbf{NP}$*-complete) if and only if its full idempotent reduct is tractable (*$\mathbf{NP}$*-complete).*

For an individual algebra $\mathbf{A} = \langle A, \mathcal{F}\rangle$, the set Inv($\mathcal{F}$) of invariant relations on A coincides with $\mathsf{SP}_{\text{fin}}(\mathbf{A})$, the set of subalgebras of finite powers of $\mathbf{A}$. We can extend this to the multi-sorted context as follows. Let $\{\mathbf{A}_i \mid i \in I\}$ be a family of finite algebras. By CSP($\{\mathbf{A}_i \mid i \in I\}$) we mean the many-sorted decision problem CSP(Γ) in which $\Gamma = \mathsf{SP}_{\text{fin}}\{\mathbf{A}_i \mid i \in I\}$ as in Definition 1.4.4. Owing to the work of Bulatov and Jeavons, we can move between many-sorted CSPs and single-sorted CSPs while preserving tractabilty by the following result.

Theorem 1.4.12 ([8, Theorem 3.4])**.** *Let* Γ *be a set of relations over the finite sets* $\{A_1, \ldots, A_n\}$*. Then there exist finite algebras* $\mathbf{A}_1, \ldots, \mathbf{A}_n$ *with universes* $A_1, \ldots, A_n$*, respectively, such that the following are equivalent:*

(a) $\mathrm{CSP}(\Gamma)$ *is tractable;*

(b) $\mathrm{CSP}(\{\mathbf{A}_1, \ldots, \mathbf{A}_n\})$ *is tractable;*

(c) $\mathbf{A}_1 \times \cdots \times \mathbf{A}_n$ *is tractable.*

A variety, $\mathcal{V}$, of algebras is said to be *tractable* if every finite algebra in $\mathcal{V}$ is tractable. The tractability of many varieties has been established by identifying special term conditions on them, or properties of the congruence lattices of algebras they contain. As a first example, following from a result of Barto and Kozik, congruence meet-semidistributivity is sufficient to establish the tractability of an algebra.

Theorem 1.4.13 ([2], Theorem 3.7)**.** *If* $\mathbf{A}$ *is a finite algebra which lies in a congruence meet-semidistributive variety, then* $\mathbf{A}$ *is tractable.*

The variety of semilattices is known to be SD($\wedge$), and is hence tractable. This was first established by Jeavons, Cohen and Gyssens [20], several years prior to the Barto and Kozik result (and did not rely on meet-semidistributivity). A finite algebra which lies in a congruence meet-semidistributive variety gives rise to a Constraint Satisfaction Problem which is solvable by the so-called "Local Consistency Method," or Bounded Width Algorithm. Larose and Zádori [29] showed that every finite, idempotent algebra which gives rise to a CSP solvable by this same method must generate a congruence meet-semidistributive variety. The Barto and Kozik result shows the converse.

The Few Subpowers Algorithm, perhaps more widely known than the Local Consistency Method, is described by the authors in [18] as the most robust "Gaussian Like" algorithm for tractable CSPs. It establishes the tractability of a finite algebra with a k-edge term, via the following result (given in [18] as Corollary 4.2).

Theorem 1.4.14. *Any finite algebra which has has, for some* $k \geq 2$*, a* k*-edge term, is tractable.*

Both Maltsev terms and near-unanimity terms give rise to k-edge terms, and thus the result of [18] subsumes those of [7] and [13].

We may connect the complexity of different algebras via the terms they possess. Given algebras $\mathbf{A} = \langle A, \mathcal{F} \rangle$ and $\mathbf{B} = \langle A, \mathcal{G} \rangle$ with the same universe and different basic operations, we say $\mathbf{A}$ and $\mathbf{B}$ are *term equivalent* if Term($\mathbf{A}$) and Term($\mathbf{B}$) contain the same nonconstant operations. Implicit in Section 3 of [9] is the following result.

Theorem 1.4.15. *If the algebras* $\mathbf{A}$ *and* $\mathbf{B}$ *are* term equivalent*, then* CSP($\mathbf{A}$) *and* CSP($\mathbf{B}$) *are the same decision problem.*

So, algebras which are term equivalent have the same complexity. We previously defined two related types of algebras: Latin squares and quasigroups. Both possess a binary multiplication which is not necessarily associative, and we can define Latin squares as specific reducts of quasigroups (or quasigroups as certain expansions of Latin squares). The class of quasigroups forms a variety, axiomatized by (1.1), and in fact, this variety has a Maltsev term, given by $q(x, y, z) = (x/(y\backslash y)) \cdot (y\backslash z)$. It follows from Theorem 1.4.14 that the variety of all quasigroups (and any of its subvarieties) is tractable.

Given a finite Latin square $\mathbf{A} = \langle A, \cdot \rangle$ of cardinality n, and any $a \in A$, the maps $L_a(x) = a \cdot x$ and $R_a(x) = x \cdot a$ are members of the symmetric group on n elements, which has order $n!$. If we define $m = n!$, $xy^2 = (xy)y$, $x^2y = x(xy)$ and inductively define $xy^{j+1} = (xy^j)y$ and $x^{j+1}y = x(x^jy)$, then the term operations $x/y = xy^{m-1}$ and $x\backslash y = x^{m-1}y$ must satisfy (1.1). Thus, every finite Latin square is term equivalent to a quasigroup, and by Theorem 1.4.15 the class of all finite Latin squares is tractable.

1.5 The Algebraic Dichotomy Conjecture

Feder and Vardi conjectured that every finite set Γ of relations is either tractable, or it is **NP**-complete. A related problem in the study of the CSP is to classify all tractable sets of relations. Through the use of universal algebra, it may be possible to both settle the dichotomy *and* classify tractable CSPs at the same time. We have already seen how, to settle the dichotomy, we may restrict our attention to decision problems of the form CSP($\mathbf{A}$), where

A is a finite, idempotent algebra. Bulatov, Jeavons and Krokhin [9] restated the Dichotomy Conjecture in terms of idempotent algebras.

Conjecture 1.5.1. *A finite idempotent algebra* **A** *is* **NP***-complete if the two-element algebra, all of whose operations are projections, is a member of* $\mathsf{HS}(\mathbf{A})$*. Otherwise it is tractable.*

The authors were able to prove this version of the dichotomy for the special case where **A** is a finite strictly simple surjective algebra. Larose and Zádori [28] recognized that a longstanding theorem due to Taylor [47] provided a term condition which could be used to restate the above conjecture.

Definition 1.5.2. A Taylor operation is an n-ary idempotent operation $t(x_1, \ldots, x_n)$ satisfying the n identities:

$$t(x_{11}, x_{12}, \ldots, x_{1n}) \approx t(y_{11}, y_{12}, \ldots, y_{1n})$$
$$t(x_{21}, x_{22}, \ldots, x_{2n}) \approx t(y_{21}, y_{22}, \ldots, y_{2n})$$
$$\vdots$$
$$t(x_{n1}, x_{n2}, \ldots, x_{1n}) \approx t(y_{11}, y_{12}, \ldots, y_{nn})$$

in which x_{ij} and y_{ij} are variables with $x_{ii} \neq y_{ii}$ for $1 \leq i, j \leq n$.

Theorem 1.5.3. *Let* **A** *be a finite idempotent algebra. The two-element algebra, all of whose operations are projections, is a member of* $\mathsf{HS}(\mathbf{A})$ *if and only if* **A** *does not possess a Taylor term operation.*

Maróti and McKenzie [35] showed that the existence of a Taylor term for a finite algebra is equivalent to the existence of a k-ary weak near-unanimity term for some k, the last step toward a universal algebraic reformulation of the dichotomy. Putting these steps together yields what is commonly referred to as the Algebraic Dichotomy Conjecture.

Conjecture 1.5.4. *A finite idempotent algebra* **A** *is* **NP***-complete if it does not possess a* k*-ary weak near-unanimity term for any* k*. Otherwise it is tractable.*

The lack of a weak near-unanimity term of any arity is sufficient for the **NP**-completeness of a finite idempotent algebra [9, Corollary 7.3]. An affirmative answer to Conjecture 1.5.4 would settle the original Dichotomy Conjecture of Feder and Vardi, while simultaneously providing a classification of all tractable algebras. A binary operation is weak near-unanimity if and only if it is commutative and idempotent, and an associative binary WNU is a semilattice operation. As we discussed previously, algebras possessing a semilattice term are known to be tractable. If the Algebraic Dichotomy Conjecture is true, then a weaker notion of semilattice operation (a commutative, idempotent binary operation) will be sufficient for the tractability of an algebra. The remainder of this thesis will focus on confirming the Algebraic Dichotomy Conjecture for the case of a finite, idempotent algebra possessing a commutative, idempotent binary operation satisfying strictly weaker conditions than associativity.

CHAPTER 2. SPECIAL SUMS OF ALGEBRAS

2.1 Płonka Sums

A similarity type of algebras is said to be *plural* if it contains no nullary operation symbols, and at least one non-unary operation symbol. Let $\mathcal{F}$ be a sequence of operation symbols, and $\rho\colon \mathcal{F} \to \mathbb{N}$ a plural similarity type. For any semilattice $\mathbf{S} = \langle S, \vee \rangle$, let $\mathbf{S}_\rho$ denote the algebra of type ρ in which, for any $f \in \mathcal{F}$ with $\rho(f) = n$, $f(x_1, x_2, \ldots, x_n) = x_1 \vee x_2 \vee \cdots \vee x_n$. $\mathbf{S}$ can be recovered from $\mathbf{S}_\rho$ by taking, for any non-unary operation symbol f, $x \vee y = f(x, y, y, \ldots, y)$. The class $\mathcal{S}\ell_\rho = \{\mathbf{S}_\rho \mid \mathbf{S} \text{ a semilattice}\}$ forms a variety term-equivalent to the variety, $\mathcal{SL}$, of semilattices. Notice that when the similarity type consists of a single binary operation, $\mathcal{S}\ell_\rho$ and $\mathcal{SL}$ coincide.

An identity is called *regular* if the same variables appear on both sides of the equals sign, and *irregular* otherwise. A variety is called regular if it is defined by regular identities. In contrast, an identity is called *strongly irregular* if it is of the form $t(x, y) \approx x$ for some binary term t in which both x and y appear. A variety is said to be strongly irregular if it satisfies a strongly regular identity. Every strongly irregular variety has an equational base consisting of a set of regular identities and a single strongly irregular identity [37, 43]. Note that most "interesting" varieties are strongly irregular, as most Maltsev conditions involve a strongly irregular identity. For example, the Maltsev condition for congruence-permutability has a ternary term $q(x, y, z)$ satisfying $q(x, y, y) \approx x$, which is a strongly irregular identity. By contrast, the variety of semilattices is regular.

The *regularization*, $\widetilde{\mathcal{V}}$, of a variety $\mathcal{V}$ is the variety defined by all regular identities that hold in $\mathcal{V}$. Equivalently, $\widetilde{\mathcal{V}} = \mathcal{V} \vee \mathcal{S}\ell_\rho$, following from the fact that $\mathcal{S}\ell_\rho$ is the class of algebras satisfying all regular identities of type ρ. If $\mathcal{V}$ is a strongly irregular variety, there is a very good structure theory for the regularization $\widetilde{\mathcal{V}}$ (due to Płonka [40, 41]), which we shall now describe.

There are several equivalent ways to think of a semilattice: as an associative, commutative, idempotent groupoid $\langle S, \vee\rangle$; as a poset $\langle S, \leq_\vee\rangle$ with ordering $x \leq_\vee y \Leftrightarrow x \vee y = y$; and as the algebra $\mathbf{S}_\rho$ of type ρ defined above.

Definition 2.1.1. Let $\langle S, \vee\rangle$ be a semilattice, $\{\mathbf{A}_s \mid s \in S\}$ a collection of algebras of plural type $\rho\colon \mathcal{F} \to \mathbb{N}$, and $\{\phi_{s,t}\colon \mathbf{A}_s \to \mathbf{A}_t \mid s \leq_\vee t\}$ a collection of homomorphisms satisfying $\phi_{s,s} = 1_{A_s}$ and $\phi_{t,u} \circ \phi_{s,t} = \phi_{s,u}$. The *Płonka sum* (over $\mathbf{S}$) of the system $\langle \mathbf{A}_s : s \in S; \phi_{s,t} : s \leq_\vee t\rangle$ is the algebra $\mathbf{A}$ of type ρ with universe $A = \biguplus\{A_s \mid s \in S\}$ and for $f \in \mathcal{F}$ a basic n-ary operation,

$$f^{\mathbf{A}}(x_1, x_2, \ldots, x_n) = f^{\mathbf{A}_s}(\phi_{s_1,s}(x_1), \phi_{s_2,s}(x_2), \ldots, \phi_{s_n,s}(x_n))$$

in which $s = s_1 \vee s_2 \vee \cdots \vee s_n$ and $x_i \in A_{s_i}$ for $1 \leq i \leq n$.

In a Płonka sum, the component algebras $\mathbf{A}_s$ (easily seen to be subalgebras of the Płonka sum $\mathbf{A}$) are known as the *Płonka fibers*, while the homomorphisms between them are called the *fiber maps*. The *canonical projection* of a Płonka sum $\mathbf{A}$ (of the system $\langle \mathbf{A}_s : s \in S; \phi_{s,t} : s \leq_\vee t\rangle$) is the map $\pi\colon \mathbf{A} \to \mathbf{S}_\rho$; $x \in A_s \mapsto s \in S$, where $\mathbf{S}_\rho$ is the member of $\mathcal{S}\ell_\rho$ derived from $\mathbf{S}$. The algebra $\mathbf{S}_\rho$ is referred to as the *semilattice replica* of the algebra $\mathbf{A}$, and the kernel of π is the *semilattice replica congruence*. Note that the congruence classes of this congruence are precisely the Płonka fibers. In some cases, a very particular Płonka sum will be useful.

Definition 2.1.2. Let $\mathbf{A}$ be any algebra and $\mathbf{S}_2 = \langle\{0, 1\}, \leq_\vee\rangle$ the two-element join semilattice. We define the algebra $\mathbf{A}^\infty$ to be the Płonka sum of the system $\langle \mathbf{A}_s : s \in S_2; \phi_{s,t} : s \leq_\vee t\rangle$, where $\mathbf{A}_0 = \mathbf{A}$, $\mathbf{A}_1$ is the trivial algebra of the same type as $\mathbf{A}$, and $\phi_{0,1}$ is the trivial homomorphism.

A comprehensive treatment of Płonka sums and other special sums of algebras is presented in [44]. We summarize just enough of the theory for our main result.

Theorem 2.1.3 (Płonka's Theorem)**.** *Let $\mathcal{V}$ be a strongly irregular variety of algebras of plural type ρ, defined by the set Σ of regular identities, together with a strongly irregular identity of the form $x \vee y \approx x$. Then the following classes of algebras coincide.*

(1) *The regularization,* $\widetilde{\mathcal{V}}$, *of* $\mathcal{V}$.

(2) *The class* $\mathsf{Pl}(\mathcal{V})$ *of Płonka sums of* $\mathcal{V}$*-algebras.*

(3) *The variety* $\overline{\mathcal{V}}$ *of algebras of type* ρ *defined by the identities* Σ *and the following identities (for* $f \in \mathcal{F}$, $\rho(f) = n$*):*

$$x \vee x \approx x \tag{P1}$$

$$(x \vee y) \vee z \approx x \vee (y \vee z) \tag{P2}$$

$$x \vee (y \vee z) \approx x \vee (z \vee y) \tag{P3}$$

$$y \vee f(x_1, x_2, \ldots, x_n) \approx y \vee x_1 \vee x_2 \vee \cdots \vee x_n \tag{P4}$$

$$f(x_1, x_2, \ldots, x_n) \vee y \approx f(x_1 \vee y, x_2 \vee y, \ldots, x_n \vee y) \tag{P5}$$

Proof. We provide a proof for the case in which ρ consists of a single binary operation. That is, the groupoid case. The proof is given in full generality in [44]. We prove the string of containments $\mathsf{Pl}(\mathcal{V}) \subseteq \widetilde{\mathcal{V}} \subseteq \overline{\mathcal{V}} \subseteq \mathsf{Pl}(\mathcal{V})$.

To see that $\mathsf{Pl}(\mathcal{V}) \subseteq \widetilde{\mathcal{V}}$, consider a groupoid $\mathbf{A}$ which is the Płonka sum over $\mathbf{S}$ of the system $\langle \mathbf{A}_s : s \in S; \phi_{s,t} : s \leq_\vee t \rangle$. The canonical projection of $\mathbf{A}$ is the semilattice $\mathbf{S}_\rho$, and if $\mathbf{A}$ satisfied an irregular identity, so would $\mathbf{S}_\rho$. But a semilattice satisfies only regular identities, so we conclude that $\mathbf{A}$ is in the regularization $\widetilde{\mathcal{V}}$ of $\mathcal{V}$.

Since $\overline{\mathcal{V}}$ is the class of models of some regular identities, and (P1)–(P5) are immediate in $\mathcal{V}$ given that $\mathcal{V}$ satisfies $x \vee y \approx x$, it follows that $\widetilde{\mathcal{V}} \subseteq \overline{\mathcal{V}}$. It remains to show that $\overline{\mathcal{V}} \subseteq \mathsf{Pl}(\mathcal{V})$. That is, every algebra in $\overline{\mathcal{V}}$ is a Płonka sum of algebras from $\mathcal{V}$. Let $\langle A, \cdot \rangle = \mathbf{A} \in \overline{\mathcal{V}}$. Define the relation σ on $\mathbf{A}$ by

$$a \,\sigma\, b \Leftrightarrow (a \vee b = a \text{ and } b \vee a = b). \tag{2.1}$$

Clearly, σ is both reflexive and symmetric. For transitivity, suppose that $a, b, c \in A$ are such that $a \,\sigma\, b$ and $b \,\sigma\, c$. Then following from (P2) and the definition of σ,

$$a \vee c = (a \vee b) \vee c = a \vee (b \vee c) = a \vee b = a$$

$$c \vee a = (c \vee b) \vee a = c \vee (b \vee a) = c \vee b = c$$

Thus, $a \,\sigma\, c$. Why is σ a congruence on $\mathbf{A}$? Suppose that $a_1 \,\sigma\, b_1$ and $a_2 \,\sigma\, b_2$. Then

$$\begin{aligned}(a_1 \cdot a_2) \vee (b_1 \cdot b_2) &\overset{(P1)}{=} (a_1 \cdot a_2) \vee (a_1 \cdot a_2) \vee (b_1 \cdot b_2)\\ &\overset{(P4)}{=} (a_1 \cdot a_2) \vee a_1 \vee a_2 \vee b_1 \vee b_2\\ &\overset{(P3)}{=} (a_1 \cdot a_2) \vee a_1 \vee b_1 \vee a_2 \vee b_2\\ &\overset{\sigma}{=} (a_1 \cdot a_2) \vee a_1 \vee a_2\\ &\overset{(P4)}{=} (a_1 \cdot a_2) \vee (a_1 \cdot a_2)\\ &\overset{(P1)}{=} (a_1 \cdot a_2).\end{aligned}$$

Similarly, $(b_1 \cdot b_2) \vee (a_1 \cdot a_2) = (b_1 \cdot b_2)$. Thus, $(a_1 \cdot a_2) \,\sigma\, (b_1 \cdot b_2)$, so σ is a congruence on $\mathbf{A}$. We now note that $\langle \mathbf{A}/\sigma, \vee \rangle$ is a semilattice, with associativity and idempotence following from (P1) and (P2), respectively. For commutativity, observe that for $a, b \in A$

$$\begin{aligned}(a \vee b) \vee (b \vee a) &\overset{(P2)}{=} a \vee (b \vee b) \vee a\\ &\overset{(P1)}{=} a \vee b \vee a\\ &\overset{(P3)}{=} a \vee a \vee b\\ &\overset{(P1)}{=} a \vee b.\end{aligned}$$

That $(b \vee a) \vee (a \vee b) = b \vee a$ follows similarly, so $(a \vee b)/\sigma = (b \vee a)/\sigma$. In a Płonka sum, the fiber maps are indexed by elements of the underlying semilattice. To simplfy the notation, we index the maps by σ-class representatives (elements of the semilattice $\mathbf{A}/\sigma$) instead. For $a/\sigma \leq_\vee b/\sigma$, we define the map $\phi_{a,b}$ by

$$\phi_{a,b} \colon a/\sigma \to b/\sigma;\, x \mapsto x \vee b.$$

To see that $\phi_{a,b}$ maps from a/σ into b/σ, notice that

$$x \,\sigma\, a \Rightarrow (x \vee b)/\sigma = x/\sigma \vee b/\sigma = a/\sigma \vee b/\sigma = b/\sigma,$$

with the last equality following from $a/\sigma \leq_\vee b/\sigma$. To see that $\phi_{a,b}$ is well-defined, suppose that $b \,\sigma\, b'$. Then by the definition of σ,

$$x \vee b = x \vee (b \vee b') = x \vee (b' \vee b) = x \vee b'.$$

(P5) is precisely the statement that $\phi_{a,b}$ is a homomorphism.

Now, we verify that $\{\phi_{a,b}\colon a/\sigma \to b/\sigma \mid a/\sigma \leq_\vee b/\sigma\}$ is a collection of homomorphisms satisfying $\phi_{a,a} = 1_{a/\sigma}$ and $\phi_{b,c} \circ \phi_{a,b} = \phi_{a,c}$. For the former, if $x\,\sigma\,a$, then $\phi_{a,a}(x) = x \vee a = x$ by the definition of σ. For the latter, if $x\,\sigma\,a$, then

$$\phi_{b,c} \circ \phi_{a,b}(x) = (x \vee b) \vee c \overset{\text{(P2)}}{=} x \vee (b \vee c) = x \vee c = \phi_{a,c}(x).$$

Finally, we are in a position to show that $\mathbf{A}$ is the Płonka sum over $\langle \mathbf{A}/\sigma, \vee\rangle$ of the system

$$\langle \mathbf{a}/\sigma : a \in A; \phi_{a,b} : a/\sigma \leq_\vee b/\sigma\rangle.$$

From the idempotence of $\vee$, the σ-classes are actually subalgebras satisfying $x \vee y \approx x$, so they are members of $\mathcal{V}$, and $\mathbf{A}$ is the disjoint union of these subalgebras. We finish by checking that multiplication using $\cdot$ in $\mathbf{A}$ coincides with the multiplication defined for the Płonka sum. So, for $a_1, a_2 \in A$ and $a = a_1 \vee a_2$,

$$\begin{aligned}
\phi_{a_1,a}(a_1) \cdot \phi_{a_2,a}(a_2) &= (a_1 \vee a) \cdot (a_2 \vee a) \\
&\overset{\text{(P5)}}{=} (a_1 \cdot a_2) \vee a \\
&= (a_1 \cdot a_2) \vee a_1 \vee a_2 \\
&\overset{\text{(P4)}}{=} (a_1 \cdot a_2) \vee (a_1 \cdot a_2) \\
&= a_1 \cdot a_2
\end{aligned}$$

□

Note that in the variety $\mathcal{V}$, the identities (P1)–(P5) defined in Theorem 2.1.3 are all direct consequences of $x \vee y \approx x$. In $\widetilde{\mathcal{V}}$, $x \vee y$ is called the *partition operation*, since it serves to decompose an algebra into the Płonka sum of $\mathcal{V}$-algebras.

It turns out we do not need the full strength of Płonka's Theorem for our purposes. Let $\mathbf{A}$ be an algebra possessing a binary term $x \vee y$ that satisfies (P1)–(P4). Without using (P5), the previous proof showed that equation (2.1) defines a congruence σ on $\mathbf{A}$, and that $\mathbf{A}/\sigma$ is a member of $\mathcal{S\ell}_\rho$. Such an algebra might not be a Płonka sum, since we are no longer guaranteed the existence of fiber maps between congruence classes, defined in the proof of Płonka's Theorem by $a/\sigma \to b/\sigma; x \mapsto x \vee b$. This is a homomorphism precisely when equation (P5) is satisfied.

$\cdot$	0	1	2	3	4	5
0	0	0	0	4	5	4
1		1	3	2	5	4
2			2	1	5	4
3				3	0	5
4					4	0
5						5

Figure 2.1 Table for Example 2.1.5

Definition 2.1.4. We call a binary term $x \vee y$ satisfying the identities (P1)–(P4) in Theorem 2.1.3 a *pseudopartition operation.*

Let $x \vee y$ be a pseudopartition operation on $\mathbf{A}$. For any n-ary basic operation f (and hence any term), we have

$$f(x_1, \dots, x_n) \in (x_1/\sigma \vee \cdots \vee x_n/\sigma) = (x_1 \vee \cdots \vee x_n)/\sigma$$

as

$$f(x_1, \dots, x_n) \vee (x_1 \vee \cdots \vee x_n) \approx f(x_1, \dots, x_n) \vee f(x_1, \dots, x_n) \approx f(x_1, \dots, x_n)$$

and

$$(x_1 \vee \cdots \vee x_n) \vee f(x_1, \dots, x_n) \approx (x_1 \vee \cdots \vee x_n) \vee (x_1 \vee \cdots \vee x_n) \approx (x_1 \vee \cdots \vee x_n).$$

In particular, every σ-class is a subalgebra of $\mathbf{A}$. We conclude the section with an example of an algebra with a pseudopartition operation that is *not* a partition operation

Example 2.1.5. Consider the groupoid in Figure 2.1. The term $x \vee y = y(x \cdot y)$ satisfies identities (P1)–(P4). The semilattice replica congruence σ partitions this algebra into two congruence classes: $\{0, 4, 5\}$ and $\{1, 2, 3\}$. Since $0 \vee 1 = 1(0 \cdot 1) = 0$, $1/\sigma \leq_\vee 0/\sigma$. The map $\phi_{1,0} : 1/\sigma \to 0/\sigma$ is uniquely defined, however

$$\phi_{1,0}(1 \cdot 2) = (1 \cdot 2) \vee 0 = 3 \vee 0 = 0(3 \cdot 0) = 0 \cdot 4 = 5,$$

and

$$\phi_{1,0}(1) \cdot \phi_{1,0}(2) = [1 \vee 0] \cdot [2 \vee 0] = [0(1 \cdot 0)] \cdot [0(2 \cdot 0)] = 0 \cdot 0 = 0,$$

so $\phi_{1,0}$ is not a homomorphism.

2.2 Main Theorem

Theorem 2.2.1. *Let* $\mathbf{A}$ *be a finite idempotent algebra with pseudopartition operation* $x \vee y$*, such that every block of its semilattice replica congruence lies in the same tractable variety. Then* $\mathrm{CSP}(\mathbf{A})$ *is tractable.*

Proof. Let $\mathbf{A}$ be a finite idempotent algebra with pseudopartition operation $x \vee y$, and corresponding semilattice replica congruence σ. As we observed in the proof of Theorem 2.1.3, each Płonka fiber, $\mathbf{A}_a = a/\sigma$, for $a \in A$, is a subalgebra of $\mathbf{A}$.

Let $\mathcal{R} = (V, A, \mathcal{C} = \{(S_i, R_i) \mid i = 1, \ldots, n\})$ be an instance of $\mathrm{CSP}(\mathbf{A})$. We shall define an instance

$$\mathcal{T} = (V, \{\mathbf{A}_a \mid a \in A\}, \delta \colon V \to A; v \mapsto a_v, \mathcal{C}' = \{(S_i, T_i) \mid i = 1, \ldots, n\})$$

of the multisorted $\mathrm{CSP}(\{\mathbf{A}_a \mid a \in A\})$, and reduce $\mathcal{R}$ to $\mathcal{T}$. By Theorem 1.4.12, the tractability of $\mathrm{CSP}(\{\mathbf{A}_a \mid a \in A\})$ is equivalent to the tractability of $\mathrm{CSP}(\prod_{a \in A} \mathbf{A}_a)$. Since the product $\prod_{a \in A} \mathbf{A}_a$ is assumed to lie in a tractable variety, if we can reduce $\mathcal{R}$ to $\mathcal{T}$, then our original problem, $\mathrm{CSP}(\mathbf{A})$, will be tractable.

First, we define the missing pieces of the instance $\mathcal{T}$. Let $1 \leq i \leq n$. Then S_i has the form $(v_1, \ldots, v_{m_i})$, where each v_j is an element of V. For a variable $v \in V$, we shall write $v \in S_i$ to indicate that $v = v_j$ for some $j \leq m_i$. Moreover, when this occurs, $\pi_v(R_i)$ will denote the projection of R_i onto the j^{th} coordinate.

For $v \in V$, define $J_v = \{i \leq n \mid v \in S_i\}$ and set

$$B_v = \bigcap_{i \in J_v} \pi_v(R_i).$$

Since each R_i is an invariant relation on $\mathbf{A}$, B_v is a subuniverse of $\mathbf{A}$. It is easy to see that if f is a solution to $\mathcal{R}$ then $f(v) \in B_v$. Consequently, we can assume without loss of generality that each R_i is a subdirect product of $\prod_{v \in S_i} B_v$.

We define the element $a_v = \bigvee B_v$, applying the term $\vee$ to take the join of the entire set B_v. In principle, the order matters (since we are not assuming that $\vee$ is commutative), however as a consequence of the definition of a pseudopartition operation, the result will always be in the same σ-class regardless of order. We define $B'_v = \mathbf{A}_{a_v} = a_v/\sigma$. Since $B_v \leq \mathbf{A}$, we have that $a_v \in B_v \cap B'_v$. For $i = 1, \ldots, n$, with $S_i = (v_1, \ldots, v_{m_i})$, define $T_i = R_i \cap \left(B'_{v_1} \times \cdots \times B'_{v_{m_i}}\right)$.

Obviously, any solution to $\mathcal{T}$ is a solution to $\mathcal{R}$. We now show that any solution to $\mathcal{R}$ can be transformed into a solution to $\mathcal{T}$. Let $f\colon V \to A$ be a solution to $\mathcal{R}$, and define

$$g\colon V \to \bigcup_{a\in A} A_a; v \mapsto f(v) \vee a_v.$$

We need to show that $g(S_i) \in T_i$ and $g(v) \in A_{a_v} = a_v/\sigma$. We first claim that

$$(\forall v \in V \text{ and } b \in B_v)\ \ b \vee a_v \in a_v/\sigma.$$

To see this, observe that

$$a_v \vee (b \vee a_v) = a_v \vee b \vee \bigvee B_v = a_v \vee \bigvee B_v = a_v \vee a_v = a_v \tag{2.2}$$

and

$$(b \vee a_v) \vee a_v = b \vee (a_v \vee a_v) = b \vee a_v.$$

That $b \vee a_v \equiv a_v \pmod{\sigma}$ now follows from (2.1). Since f is a solution to $\mathcal{R}$, for any $v \in V$, $f(v) \in B_v$. From (2.2), with $b = f(v)$, we obtain $g(v) = f(v) \vee a_v \in B'_v = A_{a_v}$.

Fix an index $i \leq n$. Since each R_i is a subdirect product, for every $v \in S_i$ there is a tuple $\mathbf{r}^v \in R_i$ with $\pi_v(\mathbf{r}^v) = a_v$. Furthermore, for each $v \in S_i$,

$$\begin{aligned}
\pi_v(g(S_i)) = g(v) &= f(v) \vee a_v \\
&= f(v) \vee \bigvee B_v \\
&\overset{*}{=} f(v) \vee \bigvee B_v \vee \bigvee_{\substack{w\neq v \\ w\in S_i}} \pi_v(\mathbf{r}^w) \\
&= f(v) \vee a_v \vee \bigvee_{\substack{w\neq v \\ w\in S_i}} \pi_v(\mathbf{r}^w) \\
&= f(v) \vee \bigvee_{w\in S_i} \pi_v(\mathbf{r}^w).
\end{aligned}$$

The starred equality follows from (P1)–(P3) and $\pi_v(\mathbf{r}^\mathbf{w}) \in B_v$. The above allows us to conclude that $g(S_i) = f(S_i) \vee \bigvee_{w\in S_i} \mathbf{r}^w \in R_i \cap \prod_{v\in S_i} B'_v = T_i$, so g is a solution to $\mathcal{T}$, which completes the proof. $\square$

Corollary 2.2.2. *Let $\mathcal{V}$ be an idempotent, tractable variety. Then $\widetilde{\mathcal{V}}$ is a tractable variety.*

Proof. Suppose that $\mathcal{V}$ is idempotent and tractable. If $\mathcal{V}$ is regular, then $\mathcal{V} = \widetilde{\mathcal{V}}$ so there is nothing to prove. It is easy to see that an idempotent, irregular variety is strongly irregular. The claim now follows from Theorems 2.1.3 and 2.2.1. □

CHAPTER 3. BOL-MOUFANG GROUPOIDS

3.1 Definitions

We call $\mathbf{B} = \langle B, \cdot \rangle$ a *CI-groupoid* if "$\cdot$" is a commutative and idempotent binary operation. Typically, we will omit the $\cdot$ and indicate multiplication in a groupoid by juxtaposition. Let $\mathcal{C}$ stand for the variety of all CI-groupoids. A groupoid identity $p \approx q$ is of *Bol-Moufang type* if:

(i) the same 3 variables appear in p and q,

(ii) one of the variables appears twice in both p and q,

(iii) the remaining two variables appear once in each of p and q,

(iv) the variables appear in the same order in p and q.

One example is the Moufang law $x(y(zy)) \approx ((xy)z)y$. There are 60 such identities, and a systematic notation for them was introduced in [38, 39]. A variety of CI-groupoids is said to be of *Bol-Moufang type* if it is defined by one additional identity of Bol-Moufang type. We say that two identities are *equivalent* if they determine the same subvariety, relative to some underlying variety. In what follows, the underlying variety is taken to be $\mathcal{C}$. Phillips and Vojtěchovský studied the equivalence of Bol-Moufang identities relative to the varieties of loops and quasigroups, requiring the binary operation appearing in a Bol-Moufang identity to be the underlying multiplication. Akhtar and his coauthors [1] classified Bol-Moufang identities involving the left or right division operation in quasigroups and loops.

Let $p \approx q$ be an identity of Bol-Moufang type with x, y, and z the only variables appearing in p and q. Since the variables must appear in the same order in p and q, we can assume without loss of generality that they are alphabetical in order of first occurrence. There are exactly 6 ways in which the x, y, and z can form such a word of length 4, and there are exactly 5 ways in which a word of length 4 can be bracketed, namely:

A	$xxyz$	1	$o(o(oo))$
B	$xyxz$	2	$o((oo)o)$
C	$xyyz$	3	$(oo)(oo)$
D	$xyzx$	4	$(o(oo))o$
E	$xyzy$	5	$((oo)o)o$
F	$xyzz$		

If X is one of A, B, C, D, E or F, and $1 \leq i < j \leq 5$, let Xij be the identity whose variables are ordered according to X, whose left-hand side is bracketed according to i, and whose right-hand side is bracketed according to j. For instance, $E15$ [i.e. $x(y(zy)) \approx ((xy)z)y$] is (one version of) the Moufang law. Following from our previous remarks, any identity of Bol-Moufang type can be transformed into some identity Xij by renaming the variables and possibly interchanging the left- and right-hand sides. There are therefore $6 \cdot (4+3+2+1) = 60$ distinct nontrivial identities of Bol-Moufang type.

Define the operation $\cdot^{\mathrm{op}}$ by $x \cdot^{\mathrm{op}} y = y \cdot x$. The *dual* p' of a groupoid term p is the result of replacing all occurrences of $\cdot$ in p with $\cdot^{\mathrm{op}}$. The dual of a groupoid identity $p \approx q$ is the identity $q' \approx p'$. This notion of duality is consistent with the one given in [38]. As an example, the dual of the Moufang law $x(y(zy)) \approx ((xy)z)y$ is the identity $y(z(yx)) \approx ((yz)y)x$. By renaming variables, we can rewrite this as $x(y(xz)) \approx ((xy)x)z$, identified as $B15$ using the systematic notation above. One can easily identify the dual of any identity Xij of Bol-Moufang type with the identity $X'j'i'$ of Bol-Moufang type computed by the rules:

$$A' = F, \quad B' = E, \quad C' = C, \quad D' = D, \quad 1' = 5, \quad 2' = 4, \quad 3' = 3.$$

We will indicate the dual of Xij by $(Xij)'$, and call an identity Xij of Bol-Moufang type *self-dual* if Xij and $(Xij)'$ are equal.

In the following sections we explore the varieties of CI-groupoids of Bol-Moufang type. The analysis consists of a mix of equational derivation, display of counterexamples, and application

Table 3.1 Varieties of CI-groupoids of Bol-Moufang type.

Name	Equivalent Identities
$\mathcal{C}$	$B45$, $D24$, $E12$
$2\mathcal{SL}$	$A13$, $A45$, $C12$, $C45$, $F12$, $F35$
$\mathcal{X}$	$A24$, $A25$, $B24$, $B25$, $E14$, $E24$, $F14$, $F24$
$\mathcal{SL}$	$A12$, $A15$, $A23$, $A34$, $A35$, $B14$, $B15$, $B34$, $B35$, $C13$, $C14$, $C23$, $C24$, $C25$, $C34$, $C35$, $D12$, $D14$, $D23$, $D25$, $D34$, $D45$, $E13$, $E15$, $E23$, $E25$, $F13$, $F15$, $F23$, $F34$, $F45$
$\mathcal{T}_2$	$C15$
$\mathcal{T}_1$	$A14$, $F25$
$\mathcal{S}_2$	$B12$, $D15$, $E45$
$\mathcal{S}_1$	$B13$, $B23$, $D13$, $D35$, $E34$, $E35$

of Maltsev conditions. This work was greatly aided by two software packages: Prover9 / Mace4 [36] and the Universal Algebra Calculator [15]. (For a discussion of their use, see Appendix A.)

Most of the implications among the equations were first discovered using Prover9. However, this software produces derivations that are only barely human-readable. We found that it took considerable effort to rewrite the proofs to be accessible to an average reader. Because of their length, some of these derivations have been relegated to an appendix.

Examples were produced by Mace4. As a rule it is a simple matter to read the Cayley table for a binary operation and verify the witnesses to an inequation. Finally, the Universal Algebra Calculator was very useful for computing congruences and searching for Maltsev conditions that hold in particular finite algebras.

3.2 Equivalences

Before we can classify the complexity of the CSP corresponding to varieties of CI-groupoids of Bol-Moufang type, it is necessary to determine which of the identities are equivalent. After determining the distinct varieties, we will establish the tractability of several using known tools. A summary of the equivalences is given in Table 3.1. We begin with an observation that will shorten the proofs considerably.

Remark 3.2.1. For commutative groupoids, each identity of Bol-Moufang type is equivalent to its dual. In fact, for any term p in a commutative groupoid, $p' \approx p$ holds.

Theorem 3.2.2. *The Bol-Moufang identities $A14$ and $F25$ are equivalent, defining the variety we call $\mathcal{T}_1$.*

Proof. Follows immediately since $F25 = (A14)'$. □

Remarkably, $C15$ is not equivalent to any other identity of Bol-Moufang type.

Theorem 3.2.3. *The identity $C15$ is self-dual, and defines the variety we call $\mathcal{T}_2$.*

Many of the below equivalences follow without the use of all of our assumptions, which may be worth investigating further. An additional remark justifies the study of Bol-Moufang identities as generalizations of associativity, and will prove useful in a few of the theorems.

Remark 3.2.4. In any groupoid, associativity implies each identity of Bol-Moufang type.

Theorem 3.2.5. *The following Bol-Moufang identities are pairwise equivalent, and determine the variety $\mathcal{S}_1$: $B13$, $B23$, $D13$, $D35$, $E34$, $E35$.*

Proof. $B13$ and $D13$ are equivalent by commuting the last two variables. To see that $B13$ and $B23$ are equivalent, interchange the roles of y and z, and apply commutativity. The remaining three identities are dual to the others. □

Theorem 3.2.6. *The following Bol-Moufang identities are pairwise equivalent, and determine the variety $\mathcal{S}_2$: $B12$, $D15$, $E45$.*

Proof. $B12$ $[x(y(xz)) \approx x((yx)z)]$ and $D15$ $[x(y(zx)) \approx ((xy)z)x]$ are equivalent under commutativity alone. $D15$ is self-dual, while $E45$ is the dual of $B12$. □

In [6], Bulatov proved the tractability of the variety of *2-semilattices*, those groupoids satisfying all two-variable semilattice identities. In particular, this class is axiomatized by commutativity, idempotence, and the *2-semilattice law*: $x(xy) \approx xy$.

Theorem 3.2.7. *The following Bol-Moufang identities are equivalent to the 2-semilattice law, and determine the variety $2\mathcal{SL}$: $A13$, $A45$, $C12$, $C45$, $F12$, $F35$.*

Proof. The 2-semilattice law, together with idempotence, implies each of the listed identities. To see how the 2-semilattice law follows from the given identities, a few easy observations are all that is needed. For $A13$ $[x(x(yz)) \approx (xx)(yz)]$, replace z with y and complete the derivation using idempotence. For $A45$ $[(x(xy))z \approx ((xx)y)z]$:

$$\begin{aligned} x(xy) &\approx (x(xy))(x(xy)) \approx ((xx)y)(x(xy)) \\ &\approx (xy)(x(xy)) \approx (x(xy))(xy) \\ &\approx ((xx)y)(xy) \approx (xy)(xy) \approx (xy). \end{aligned}$$

For $C12$ $[x(y(yz)) \approx x((yy)z)]$:

$$\begin{aligned} x(xy) &\approx (x(xy))(x(xy)) \approx (x(xy))((xx)y) \\ &\approx (x(xy))(xy) \approx (xy)(x(xy)) \\ &\approx (xy)((xx)y) \approx (xy)(xy) \approx (xy). \end{aligned}$$

The remainder of the identities are dual to those investigated, so it follows from Remark 3.2.1 that they each imply the 2-semilattice law. □

The following lemmas will aid in proving the largest groups of equivalences.

Lemma 3.2.8. *Each of following Bol-Moufang identities, together with idempotence, implies the 2-semilattice law:* $A24$, $A25$, $A34$, $B35$, $C35$, $D23$.

Proof. For $A24$ $[x((xy)z) \approx (x(xy))z]$:

$$x(xy) \approx x((xx)y) \approx (x(xx))y \approx (xx)y \approx xy.$$

For $A25$ $[x((xy)z) \approx ((xx)y)z]$:

$$x(xy) \approx x((xy)(xy)) \approx ((xx)y)(xy) \approx (xy)(xy) \approx xy.$$

For $A34$ $[(xx)(yz) \approx (x(xy))z]$:

$$x(xy) \approx (xx)(xy) \approx (x(xx))y \approx xy.$$

For $B35$ $[(xy)(xz) \approx ((xy)x)z]$ and $C35$ $[(xy)(yz) \approx ((xy)y)z]$:

$$x(xy) \approx (xx)(xy) \approx ((xx)x)y \approx xy.$$

	0	1	2
0	0	2	1
1	0	1	2
2	0	1	2

(a) Example 3.2.10

	0	1	2
0	0	2	1
1	1	1	1
2	2	2	2

(b) Example 3.2.11

Figure 3.1 Tables for Examples 3.2.10 and 3.2.11

For $D23$ $[x((yz)x) \approx (xy)(zx)]$:

$$x(xy) \approx x(yx) \approx x((yy)x) \approx (xy)(yx) \approx (xy)(xy) \approx xy. \quad \square$$

Lemma 3.2.9. *Each of the following Bol-Moufang identities, together with commutativity and idempotence, implies the 2-semilattice law:* $A15$*,* $A23$*,* $B14$*,* $C14$*.*

Proof. For $A15$ $[x(x(yz)) \approx ((xx)y)z]$:

$$\begin{aligned} x(xy) &\approx (xy)x \approx ((xx)y)x \approx x(x(yx)) \approx x(x(xy)) \approx x(x(x(yy))) \\ &\approx x(((xx)y)y) \approx x((xy)y) \approx ((yx)y)x \approx (((yx)(yx))y)x \\ &\approx (yx)((yx)(yx)) \approx yx \approx xy. \end{aligned}$$

For $A23$ $[x((xy)z) \approx (xx)(yz)]$:

$$x(xy) \approx x((xy)(xy)) \approx (xx)(y(xy)) \approx x(y(xy)) \approx x((xy)y) \approx (xx)(yy) \approx xy.$$

For $B14$ $[x(y(xz)) \approx (x(yx))z]$:

$$x(xy) \approx x(yx) \approx x(y(xx)) \approx (x(yx))x \approx x(x(xy)) \approx (x(xx))y \approx xy.$$

For $C14$ $[x(y(yz)) \approx (x(yy))z]$:

$$x(xy) \approx (yx)x \approx (y(xx))x \approx y(x(xx)) \approx yx \approx xy. \quad \square$$

Example 3.2.10. Figure 3.1(a) is an idempotent groupoid satisfying $A15$ and $A23$ which does not satisfy the 2-semilattice law (it fails since $0(0 \cdot 1) \neq 0 \cdot 1$).

Example 3.2.11. Figure 3.1(b) is an idempotent groupoid satisfying $B14$ and $C14$ which does not satisfy the 2-semilattice law (it fails since $0(0 \cdot 1) \neq 0 \cdot 1$).

Lemma 3.2.12. *$F45$, together with commutativity and idempotence, implies the 2-semilattice law.*

Proof. $F45$ $[(x(yz))z \approx ((xy)z)z]$ commutes to become $z((xy)z) \approx z(x(yz))$. A few intermediate identities:

1. $(xy)(x(y(xy))) \approx xy$ follows by replacing z with xy in the commuted version of $F45$.
2. $(yx)x \approx x(y(yx))$ follows by replacing x with y, and z with x in the commuted $F45$.
3. $x(yx) \approx x(y(xy))$ is just the previous identity with commutativity applied.
4. $(xy)(x(yx)) \approx xy$ follows from (1) and (3) above.

We now have enough for the 2-semilattice law:

$$\begin{aligned} x(xy) &\approx x(yx) \approx [x(yx)][x(yx)] \\ &\approx [x(yx)][x(y(xy))] \\ &\approx [x(yx)][x(y(x(yx)))] \\ &\approx [x(yx)][xy] \approx [xy][x(yx)] \approx xy. \end{aligned}$$

□

Several of the identities in Lemmas 3.2.8 and 3.2.9 determine a subvariety of $\mathcal{C}$ consisting of 2-semilattices. However, as nothing further was known about this subvariety as of this writing, we give it the name $\mathcal{X}$.

Theorem 3.2.13. *The following Bol-Moufang identities are pairwise equivalent, and determine the variety $\mathcal{X}$, a subvariety of 2-semilattices: $A24$, $A25$, $B24$, $B25$, $E14$, $E24$, $F14$, $F24$.*

Proof. The identities $A24$ and $B24$ are easily seen to be equivalent by commuting the variables in the innermost set of parentheses. $A25$ and $B25$ are equivalent in the same way. We will show that $A24$ and $A25$ are equivalent, with the help of Lemma 3.2.8. To see that $A24$ implies $A25$, observe that $((xx)y)z \approx (xy)z \approx (x(xy))z \approx x((xy)z)$. Conversely, from $A25$ we can derive $x((xy)z) \approx ((xx)y)z \approx (xy)z \approx (x(xy))z$. The remaining identities are dual to those investigated. □

Theorem 3.2.14. *Each of the following Bol-Moufang identities is equivalent to associativity, and determines the variety $\mathcal{SL}$ of semilattices: $A12$, $A15$, $A23$, $A34$, $A35$, $B14$, $B15$, $B34$, $B35$, $C13$, $C14$, $C23$, $C24$, $C25$, $C34$, $C35$, $D12$, $D14$, $D23$, $D25$, $D34$, $D45$, $E13$, $E15$, $E23$, $E25$, $F13$, $F15$, $F23$, $F34$, $F45$.*

Proof. We proceed via a few closed loops of equivalences. Wherever the 2-semilattice law is used, it has already been proven to hold in Lemma 3.2.8, Lemma 3.2.9, or Lemma 3.2.12. Associativity implies any of the listed identities by our previous remark.

- $A23 \Rightarrow D12 \Rightarrow D14 \Rightarrow F45 \Rightarrow F34 \Rightarrow A23$
  - $A23 \Rightarrow D12$:

$$x(y(zx)) \approx x((xz)y) \approx (xx)(zy) \approx x(zy) \approx x(x(zy)) \approx x((yz)x)$$

  - $D12$ and $D14$ are equivalent under commutativity.
  - $D12 \Rightarrow F45$:

$$(x(yz))z \approx z(x(yz)) \approx z((xy)z) \approx ((xy)z)z$$

  - $F45 \Rightarrow F34$:

$$(xy)(zz) \approx (xy)z \approx ((xy)z)z$$

  - $F34$ is the dual of $A23$.

- $A23 \Rightarrow C35 \Rightarrow C34 \Rightarrow$ Associativity $\Rightarrow A34 \Rightarrow$ Associativity $\Rightarrow A23$
  - $A23 \Rightarrow C35$:

$$\begin{aligned}(xy)(yz) &\approx [(xy)(xy)](yz) \approx (xy)[((xy)y)z] \\ &\approx (xy)((xy)z) \approx (xy)z \approx ((xy)y)z\end{aligned}$$

  - $C35 \Rightarrow C34$:

$$(xy)(yz) \approx ((xy)z)z \approx (xy)z \approx (x(yy))z$$

  - $C34 \Rightarrow$ Associativity:

$$(xy)z \approx (x(yy))z \approx (xy)(yz) \approx (zy)(yx) \approx (z(yy))x \approx (zy)x \approx x(yz)$$

- $A34 \Rightarrow$ Associativity:

$$x(yz) \approx (xx)(yz) \approx (x(xy))z \approx (xy)z$$

- $C35 \Rightarrow B35 \Rightarrow D23 \Rightarrow C14 \Rightarrow A15 \Rightarrow C34$

  - $C35 \Rightarrow B35$:

$$(xy)(xz) \approx (yx)(xz) \approx ((yx)x)z \approx ((xy)x)z$$

  - $B35 \Rightarrow D23$:

$$\begin{aligned} x((yz)x) &\approx x(yz) \approx (yz)x \approx (yz)(yx) \approx (yx)(yz) \approx ((yx)y)z \\ &\approx (yx)z \approx (xy)z \approx ((xy)x)z \approx (xy)(xz) \approx (xy)(zx) \end{aligned}$$

  - $D23 \Rightarrow C14$:

$$\begin{aligned} x(y(yz)) &\approx x(yz) \approx x((yz)x) \approx (xy)(zx) \\ &\approx (yx)(xz) \approx (yx)[(xz)(yx)] \approx [(yx)x][z(yx)] \\ &\approx [yx][z(yx)] \approx z(yx) \approx (xy)z \approx (x(yy))z \end{aligned}$$

  - $C14 \Rightarrow A15$:

$$x(x(yz)) \approx x(yz) \approx x(y(yz)) \approx (x(yy))z \approx (xy)z \approx ((xx)y)z$$

  - $A15 \Rightarrow C34$:

$$(xy)(yz) \approx (xy)((xy)(yz)) \approx (((xy)(xy))y)z \approx ((xy)y)z \approx (xy)z \approx (x(yy))z$$

- $B35 \Leftrightarrow B14 \Leftrightarrow B15$

  - $B35 \Rightarrow B14$:

    1. $B35$ simplifies to $(xy)(xz) \approx (xy)z$ under the 2-semilattice law.
    2. $(xy)z \approx (xz)y$ follows by permuting the variables in the left hand side of the above.
    3. $x(yz) \approx z(xy)$ follows by permuting the variables in the above, and applying commutativity.

4. Lastly, using the previous equation with xz substituted for z yields $x(y(xz)) \approx (xz)(xy) = (xy)z \approx (x(xy))z \approx (x(yx))z$, which is $B14$.

– $B14 \Rightarrow B35$:

$$(xy)(xz) \approx (yx)(xz) \approx (x(yx))(xz) \approx x(y(x(xz)))$$
$$\approx x(y(xz)) \approx (x(yx))z \approx ((xy)x)z$$

– $B14$ and $B15$ are equivalent under commutativity.

Applying idempotence, one can derive associativity from $A35$ $[(xx)(yz) \approx ((xx)y)z]$ or $C24$ $[x((yy)z) \approx (x(yy))z]$, and so both are equivalent to associativity. $B34$ $[(xy)(xz) \approx (x(yx))z]$ and $B35$ $[(xy)(xz) \approx ((xy)x)z]$ are equivalent under commutativity. The remaining identities are dual to those investigated. □

There is one last class of equivalent identities of Bol-Moufang type. It is in some sense trivial.

Theorem 3.2.15. *The identities $B45$ $[(x(yx))z \approx ((xy)x)z]$, $D24$ $[x((yz)x) \approx (x(yz))x]$, and $E12$ $[x(y(zy)) \approx x((yz)y)]$ are equivalent, and determine the variety $\mathcal{C}$.*

Proof. It is easy to see that all three identities follow immediately from commutativity. □

It is worth noting that although any one of $B45$, $D24$, or $E12$ defines the entire variety of CI-groupoids, they do not guarantee commutativity, even in the presence of idempotence.

Example 3.2.16. A two element left-zero semigroup satisfies $B45$, $D24$, and $E12$, but is not commutative.

3.3 Implications

We now show how the 8 varieties of CI-groupoids of Bol-Moufang type are related.

Theorem 3.3.1. *The following inclusions hold among the varieties of CI-groupoids of Bol-Moufang type: $\mathcal{SL} \subseteq \mathcal{X} \subseteq 2\mathcal{SL} \subseteq \mathcal{C}$, $\mathcal{SL} \subseteq \mathcal{T}_1 \subseteq \mathcal{T}_2 \subseteq \mathcal{C}$, $\mathcal{SL} \subseteq \mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{C}$.*

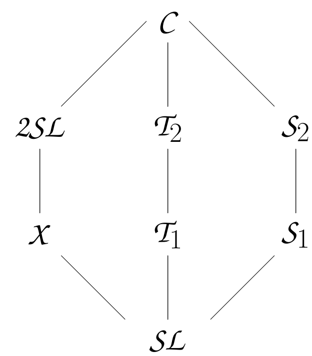

Figure 3.2 Varieties of CI-groupoids of Bol-Moufang Type

Proof. The variety $\mathcal{SL}$ of semilattices is contained in all the others, following from Remark 3.2.4. Likewise, they are all trivially contained in $\mathcal{C}$. To see that $\mathcal{X}$ is contained in $2\mathcal{SL}$, note that in the proof of Lemma 3.2.8, we showed that both $A24$ and $A25$, which define the variety $\mathcal{X}$, imply the 2-semilattice law. To see that $\mathcal{T}_1 \subseteq \mathcal{T}_2$, we show that $A14$ $[x(x(yz)) \approx (x(xy))z]$ implies $C15$ $[x(y(yz)) \approx ((xy)y)z]$. Assuming $A14$, we have: $x(y(yz)) \approx (y(yz))x \approx y(y(zx)) \approx y(y(xz)) \approx (y(yx))z \approx ((xy)y)z$. Lastly, to see that $\mathcal{S}_1 \subseteq \mathcal{S}_2$, we show that $B13$ $[x(y(xz)) \approx (xy)(xz)]$ implies $B12$ $[x(y(xz)) \approx x((yx)z)]$. Assuming $B13$, we have $x(y(xz)) \approx (xy)(xz) \approx (xz)(xy) \approx x(z(xy)) \approx x((yx)z)$. □

A Hasse diagram of the situation (with inclusions directed upward, so that higher varieties are larger) is shown in Figure 3.2. Up to this point, we have justified only the inclusions, but we must still show that they are *proper*, and that no inclusions have been missed.

3.4 Distinguishing Examples

We now show that the 8 varieties of CI-groupoids of Bol-Moufang type are distinct. We have aimed to use as few examples as possible. While the 7 groupoids presented suffice to show that all inclusions are proper, there may be some larger groupoids which subsume multiple examples. For readability, and since each example is commutative, only the upper triangle of each Cayley table is given.

	0	1	2
0	0	2	1
1		1	1
2			2

(a) Example 3.4.1

	0	1	2
0	0	1	0
1		1	2
2			2

(b) Example 3.4.2

Figure 3.3 Tables for Examples 3.4.1 and 3.4.2

	0	1	2	3
0	0	3	2	3
1		1	2	3
2			2	3
3				3

(a) Example 3.4.3

	0	1	2	3	4	5
0	0	0	0	4	5	4
1		1	3	2	5	4
2			2	1	5	4
3				3	0	5
4					4	0
5						5

(b) Example 3.4.4

Figure 3.4 Tables for Examples 3.4.3 and 3.4.4

Example 3.4.1. Figure 3.3(a) is a CI-groupoid which is not in $\mathcal{2SL} \cup \mathcal{T}_2 \cup \mathcal{S}_2$. The 2-semilattice law fails because $0(0 \cdot 1) \neq 0 \cdot 1$; $C15$ fails because $0(1(1 \cdot 1)) \neq ((0 \cdot 1)1)1$; $B12$ fails because $0(0(0 \cdot 1)) \neq 0((0 \cdot 0)1)$.

Example 3.4.2. Figure 3.3(b) is a 2-semilattice which is not in $\mathcal{X}$. $A24$ fails because $0((0 \cdot 1)2) \neq (0(0 \cdot 1))2$.

Example 3.4.3. Figure 3.4(a) is member of $\mathcal{X}$ which is not a member of $\mathcal{T}_2$ or $\mathcal{S}_2$, and is also not a semilattice. $C15$ fails because $0(1(1 \cdot 2)) \neq ((0 \cdot 1)1)2$. $B12$ fails because $0(1(0 \cdot 2)) \neq 0((1 \cdot 0)2)$. Associativity fails because $(0 \cdot 1)2 \neq 0(1 \cdot 2)$.

Example 3.4.4. Figure 3.4(b) is a member of $\mathcal{T}_2$ which is not in $\mathcal{T}_1$. $A14$ fails because $0(0(1 \cdot 2)) \neq (0(0 \cdot 1))2$.

Example 3.4.5. Figure 3.5(a) is member of $\mathcal{T}_1$ which is neither a 2-semilattice, nor a member of $\mathcal{S}_2$, and hence is not a semilattice. The 2-semilattice law fails because $0(0 \cdot 1) \neq (0 \cdot 0)1$, while $B12$ fails because $0(0(0 \cdot 1)) \neq 0((0 \cdot 0)1)$.

Example 3.4.6. Figure 3.5(b) is a member of $\mathcal{S}_2$ which is not a member of $\mathcal{S}_1$. $B13$ fails because $0(1(0 \cdot 1)) \neq (0 \cdot 1)(0 \cdot 1)$.

	0	1	2
0	0	2	1
1		1	0
2			2

(a) Example 3.4.5

	0	1	2	3
0	0	2	3	3
1		1	3	3
2			2	3
3				3

(b) Example 3.4.6

	0	1	2
0	0	2	0
1		1	1
2			2

(c) Example 3.4.7

Figure 3.5 Tables for Examples 3.4.5, 3.4.6 and 3.4.7

Example 3.4.7. Figure 3.5(c) is a member of $\mathcal{S}_1$ which is neither a 2-semilattice, nor a member of $\mathcal{T}_2$, and hence is not a semilattice. The 2-semilattice law fails because $0(0 \cdot 1) \neq 0 \cdot 1$, while $C15$ fails because $0(0(0 \cdot 1)) \neq ((0 \cdot 0)0)1$.

While the Hasse diagram presented in Figure 3.2 is not likely to be a lattice, we note that all of the intersections are true — that is, $\mathcal{2SL} \cap \mathcal{T}_2 = \mathcal{2SL} \cap \mathcal{S}_2 = \mathcal{T}_2 \cap \mathcal{S}_2 = \mathcal{SL}$.

3.5 Properties of Bol-Moufang CI-Groupoids

Our analysis thus far has determined properties of several, but not all of the varieties of CI-groupoids of Bol-Moufang type. In Theorem 3.2.7 we showed that each of the listed identities was equivalent to the 2-semilattice law. Since $\mathcal{X}$ is a subvariety of $\mathcal{2SL}$, it is also a variety of 2-semilattices. Likewise, we showed in Theorem 3.2.14 that all of the listed identities are equivalent to the associative law, and thus determine the variety of semilattices. Following from the result of Bulatov [6], we know all three of these varieties ($\mathcal{SL}$, $\mathcal{2SL}$, and $\mathcal{X}$) to be tractable. That the variety $\mathcal{C}$ is indeed the variety of all CI-groupoids follows from the fact that $B45$, $D24$, $E12$ are immediate consequences of commutativity. The remainder of this section, as well as the next, is devoted to the other four varieties.

Using the Universal Algebra Calculator [15], in conjunction with Mace4 [36], we investigated Maltsev conditions satisfied by the varieties $\mathcal{T}_1$ and $\mathcal{T}_2$. With Mace4, we generated the only three element algebra in $\mathcal{T}_2 \setminus \mathcal{T}_1$ (Example 3.4.5), and provided it as input to the Universal Algebra Calculator. For this algebra, the Calculator did not find a majority, Pixley, or near-unanimity term, or terms for congruence distributivity, congruence join semi-distributivity, or congruence meet semi-distributivity. We then generated a 4-element CI-groupoid satisfying

$A14$, for which the UA Calculator found only the Taylor term $x \cdot y$, inspiring our names for $\mathcal{T}_1$ and $\mathcal{T}_2$. Since $\mathcal{SL} \subseteq \mathcal{T}_1 \subseteq \mathcal{T}_2$, these varieties are not congruence modular (and hence cannot be shown tractable via the result of [18]). Following from the result of Kearnes and Kiss (applied in the next theorem), we have that they are also not congruence meet-semidistributive, and so the Barto and Kozik result cannot be applied to $\mathcal{T}_2$ as a whole.

Theorem 3.5.1. *The variety $\mathcal{T}_1$ (and hence $\mathcal{T}_2$) is not congruence meet-semidistributive.*

Proof. By Theorem 1.3.6, it is enough to produce a variety $\mathcal{M}$ of modules, together with a family of idempotent Maltsev conditions that is satisfied in both $\mathcal{T}_1$ and $\mathcal{M}$. Consider the variety $\mathcal{M}$ of modules over the ring $\mathbb{Z}_3$. Define the term $x \cdot y = 2(x + y)$ in $\mathcal{M}$. Take as our family of idempotent Maltsev conditions the axioms defining $\mathcal{T}_1$:

$$x \cdot x \approx x$$

$$x \cdot y \approx y \cdot x$$

$$x \cdot (x \cdot (y \cdot z)) \approx (x \cdot (x \cdot y)) \cdot z.$$

In $\mathcal{M}$,

$$\begin{aligned}
x \cdot x &\approx 2(x + x) \\
&\approx 2x + 2x \\
&\approx 4x \approx x \\
x \cdot y &\approx 2(x + y) \\
&\approx 2x + 2y \\
&\approx 2y + 2x \approx y \cdot x \\
x \cdot (x \cdot (y \cdot z)) &\approx 2(x + 2(x + 2(y + z))) \\
&\approx 2x + 4x + 8y + 8z \\
&\approx 6x + 8y + 8z \\
&\approx 12x + 8y + 2z \\
&\approx 4x + 8x + 8y + 2z
\end{aligned}$$

$$\approx 2(2(x + 2(x + y)) + z)$$
$$\approx (x \cdot (x \cdot y)) \cdot z. \qquad \square$$

Theorem 3.5.2. *$\mathcal{2SL}$ is congruence meet-semidistributive.*

Proof. We wish to show that there is a family of idempotent Maltsev conditions that is satisfied in $\mathcal{2SL}$, but is only true in the trivial variety of modules. The result will then follow by Theorem 1.3.6. We take as our family the identities defining $\mathcal{2SL}$:

$$x \cdot x \approx x$$
$$x \cdot y \approx y \cdot x$$
$$x \cdot (x \cdot y) \approx x \cdot y.$$

Without loss of generality, we may consider only modules over *unital rings*, those rings R with multiplicative identity 1_R. Also, we can assume that if a variety of R-modules has some $r \in R$ such that the identity $rx \approx 0$ is satisfied, then $r = 0_R$, the additive identity element of R. From the above simplifying assumptions, we can conclude that if $r \in R$ is such that a variety of R-modules satisfies $rx \approx x$, then $r = 1_R$. Now, suppose that there is a variety $\mathcal{M}$ of R-modules which had a binary term $x \cdot y$ satisfying the idempotent and commutative laws. Any binary R-module term must have the form $rx + sy$, for some $r, s \in R$. From the commutative law, we derive that

$$rx + sy \approx x \cdot y \approx y \cdot x \approx ry + sx,$$

i.e. $(r - s)x + (s - r)y = 0$. Setting y to be the zero element of the module, we can derive the fact that $r = s$, and so the term $x \cdot y = rx + ry$ for some $r \in R$. From the idempotent law, it must be the case that

$$x \approx x \cdot x \approx rx + rx \approx (r + r)x,$$

which implies that $r + r = 1_R$.

Finally, assuming that $x \cdot y$ also satisfies the 2-semilattice law, we see that $x \cdot (x \cdot y) \approx x \cdot y$ implies that $rx + r^2x + r^2y = rx + ry$, i.e. $r^2x + (r^2 - r)y \approx 0$. Successively letting $y = 0$ and $x = 0$, we conclude that $r^2 = r = 0_R$. Returning to the idempotent law, this implies that the variety $\mathcal{M}$ satisfies the identity $x \approx x \cdot x \approx 0_R x + 0_R x \approx 0$. That is, the variety $\mathcal{M}$ must be trivial. □

Using Theorem 3.5.2 and Theorem 1.4.13 provides an alternative to Bulatov's proof of the tractability of $\mathcal{2SL}$, and also a proof of the well-known fact that $\mathcal{SL}$ (a subvariety of $\mathcal{2SL}$) is SD($\wedge$).

Theorem 3.5.3. *$\mathcal{S}_2$ is congruence meet-semidistributive.*

Proof. We wish to show that there is a family of idempotent Maltsev conditions that is satisfied in $\mathcal{S}_2$, but is only true in the trivial variety of modules. The result will then follow by Theorem 1.3.6. We take as our family the identities defining $\mathcal{S}_2$:

$$x \cdot x \approx x$$

$$x \cdot y \approx y \cdot x$$

$$x \cdot (y \cdot (x \cdot z)) \approx x \cdot ((y \cdot x) \cdot z).$$

Following from the discussion in the previous theorem, we need only consider modules over unital rings, and we may assume without loss of generality that the R-modules in question satisfy the implications $rx \approx 0 \Rightarrow r = 0_R$, $rx \approx x \Rightarrow r = 1_R$. Now, suppose that there is a variety $\mathcal{M}$ of R-modules which had a binary term $x \cdot y$ satisfying the above Maltsev conditions. Such a term must be of the form $x \cdot y \approx rx + ry$. Interpreting the final axiom using this term yields the identity

$$r(x + r(y + r(x + z))) \approx r(x + r(r(y + x) + z)).$$

Rearranging the above we derive the identity $r^2(y - z) + r^3(z - y) \approx 0$, which is equivalent to $(r^2 - r^3)(y - z) \approx 0$. Replacing z by the 0 element of the module yields $(r^2 - r^3)y \approx 0$, which implies $r^2 - r^3 = 0_R$ (equivalently, $r^2 = r^3$). A little further manipulation in the ring R allows us to show that

$$r^2 = r^3 = r^2(r) = r^2(1_R - r) = r^2 - r^3 = 0_R,$$

following from the previous observation that $r + r = 1_R$. Squaring both sides of $r + r = 1_R$ gives

$$0_R = 4r^2 = (r + r)^2 = (1_R)^2 = 1_R,$$

so the ring R is trivial, and as a result the variety $\mathcal{M}$ must satisfy $x \approx 1_R x \approx 0_R x \approx 0$. That is, $\mathcal{M}$ must be trivial. $\square$

Following immediately from Theorems 3.5.3 and 1.4.13, we have the following corollary.

Corollary 3.5.4. *$\mathcal{S}_2$ is tractable.*

3.6 The Structure of $\mathcal{T}_1$ and $\mathcal{T}_2$

Recall that $\mathcal{T}_1$ is the variety of commutative, idempotent groupoids axiomatized by the additional identity $A14$ $[x(x(yz)) \approx (x(xy))z]$. $\mathcal{T}_1$ is contained in the variety $\mathcal{T}_2$ of CI-groupoids satisfying $C15$ $[x(y(yz)) \approx ((xy)y)z]$. Recall also that xy is a Taylor term for both $\mathcal{T}_1$ and $\mathcal{T}_2$, but neither variety satisfies any familiar Maltsev conditions. As such, the Few Subpowers and Bounded Width Algorithms cannot be used to solve the CSP over an arbitrary algebra from $\mathcal{T}_1$ or $\mathcal{T}_2$. As it turns out, we may use our main result to obtain the tractability of both, and additionally we obtain a strong structure theory for $\mathcal{T}_1$. To prove that $\mathcal{T}_2$ is tractable, we need a few lemmas, following which we give a pseudopartition operation for the variety.

Lemma 3.6.1. *The variety $\mathcal{T}_2$ satisfies the following identities:*

$$x(y(yx)) \approx y(yx) \tag{3.1}$$

$$x(y(x(x(y(x(xz)))))) \approx x(y(yz)) \tag{3.2}$$

$$x(y(yz)) \approx x(y(y(x(xz)))) \tag{3.3}$$

$$(xy)(x(xz)) \approx (xy)z \tag{3.4}$$

$$x[y(y(z(zu)))] \approx x[(yz)(u(yz))] \tag{3.5}$$

$$x(y(z(z(y(z(zu)))))) \approx x(y(y(z(zu)))) \tag{3.6}$$

$$x(y(x(z(zy)))) \approx z(z(y(yx))) \tag{3.7}$$

$$x(y(y(z(y(yx))))) \approx x(z(y(yx))) \tag{3.8}$$

$$(x(y(yz)))(y(yu)) \approx (x(y(yz)))u \tag{3.9}$$

$$x(y(y(z(zx)))) \approx y(y(z(zx))) \tag{3.10}$$

$$(xy)(z(xy)) \approx y(y(x(xz))) \tag{3.11}$$

$$x(x(y(yz))) \approx y(y(x(xz))) \tag{3.12}$$

Proof. See Appendix B. □

Lemma 3.6.2. *The variety $\mathcal{T}_2$ satisfies the identity*

$$x(x(y(yz))) \approx (y(xy))(z(y(xy))). \tag{3.13}$$

Proof. See Appendix B. □

Theorem 3.6.3. *$x \vee y = y(xy)$ is a pseudopartition operation for $\mathcal{T}_2$.*

Proof. See Appendix B. □

Definition 3.6.4. A CI-groupoid satisfying $x(xy) \approx y$ is called a *squag* or *Steiner quasigroup.*

The quasigroup label is justified as the equation $ax = b$ has the unique solution $x = ab$ in any squag. Squags completely capture Steiner triple systems from combinatorics in an algebraic framework. A brief survey is presented in [11, Chapter 3], while a more detailed exploration of squags and related objects can be found [42]. As a variety of Latin squares, the variety of squags is tractable.

Corollary 3.6.5. *$\mathcal{T}_2$ is tractable.*

Proof. Let $\mathbf{A}$ be a finite member of $\mathcal{T}_2$. We showed in Theorem 3.6.3 that $x \vee y = y(xy)$ is a pseudopartition operation for $\mathcal{T}_2$. From the discussion following Theorem 2.1.3, each Płonka fiber of $\mathbf{A}$ satisfies $x \approx x \vee y \approx y(xy)$. Thus each block of the semilattice replica congruence lies in the variety of squags. Therefore, by Theorem 2.2.1, $\mathbf{A}$ is tractable. □

This completes our proof of the tractability of all varieties of CI-groupoids of Bol-Moufang type, with the exception of the variety $\mathcal{C}$ of all CI-groupoids. We can obtain a still stronger result regarding the structure of $\mathcal{T}_1$. Let $\Sigma = \{xx \approx x, xy \approx yx, x(x(yz)) \approx (x(xy))z\}$, and let $x \vee y = y(xy)$ be the pseudopartition operation for $\mathcal{T}_2$. Note that $\mathcal{T}_1 = \operatorname{Mod}(\Sigma)$. Define $\mathcal{W} = \operatorname{Mod}(\Sigma \cup \{x \vee y \approx x\})$.

As noted above, the variety of squags is the variety of CI-groupoids satisfying $x(xy) \approx x(yx) \approx y$. From the squag identity, we can easily derive $A14$: $x(x(yz)) \approx yz \approx (x(xy))z$, which immediately gives:

Lemma 3.6.6. *$\mathcal{W}$ is the variety of squags.*

We will show that $\mathcal{T}_1$ is actually the regularization of $\mathcal{W}$, following from Theorem 2.1.3, by proving that $x \vee y$ is a partition operation for $\mathcal{T}_1$.

Theorem 3.6.7. *The variety $\mathcal{T}_1$ is the regularization of the variety of squags.*

Proof. Let $\mathcal{W}$ be the variety of squags as defined above. To prove that $\mathcal{T}_1 = \widetilde{\mathcal{W}}$, it suffices to show that Σ can be used to derive each of the identities in Theorem 2.1.3(3). Since (P1)–(P4) are shown in Theorem 3.6.3, and $\mathcal{T}_1$ is a subvariety of $\mathcal{T}_2$, we need only justify identity (P5): $(xy) \vee z \approx (x \vee z)(y \vee z)$. As before, we do not label idempotence or commutativity.

$$\begin{aligned}(xy) \vee z &\approx z((xy)z) \approx z(z(yx)) \approx z((z(zz))(yx)) \\ &\overset{A14}{\approx} z(z(z(z(yx)))) \overset{A14}{\approx} z(z((z(zy))x)) \approx z(z(x(z(zy)))) \\ &\overset{A14}{\approx} (z(zx))(z(zy)) \approx (z(xz))(z(yz)) \approx (x \vee z)(y \vee z).\end{aligned}$$

□

As a consequence of this theorem, every member of $\mathcal{T}_1$ is a Płonka sum of squags. The term $x \vee y = y(xy)$ is, however, not a partition operation for $\mathcal{T}_2$. Example 3.4.4 is an algebra in $\mathcal{T}_2$ for which the given pseudopartition operation fails to satisfy (P5), and so the algebras in $\mathcal{T}_2$ need not be Płonka sums, although they will decompose as disjoint unions of squags.

CHAPTER 4. FURTHER GENERALIZATIONS

4.1 Distributive and Entropic CI-Groupoids

In the previous chapter we analyzed, as far as possible with current techniques, the tractability of the varieties of CI-groupoids of Bol-Moufang type. We continue the CSP-focused analysis of CI-groupoids by studying other weakenings of associativity.

One such identity, often studied in conjunction with commutativity and idempotence, is the distributive law $x(yz) \approx (xy)(xz)$. We will refer to the variety of commutative, idempotent distributive groupoids as the variety of *CID-groupoids*. They are, in some sense, the "end of the line" for our inquiry. In their booklet [22], summarizing the state of the art in distributive groupoids, Ježek, Kepka, and Němec share their opinion that "the deepest non-associative theory within the framework of groupoids" is the theory of distributive groupoids.

Another identity we will consider is the entropic law $(xy)(zw) \approx (xz)(yw)$. In the literature this is sometimes referred to as mediality or the abelian law. A complete description of the lattice of subvarieties of commutative, idempotent, entropic groupoids (which we will call *CIE-groupoids*) is given in [21, Theorem 4.9]. Every idempotent, entropic groupoid (and hence every CIE-groupoid) is distributive. In [25], Kepka and Němec show that every CID-groupoid which is not entropic has cardinality at least 81, so for the more general case of CID-groupoids, generating models and inspecting them for patterns is no longer a reasonable approach. Fortunately, Płonka sums again prove useful.

Theorem 4.1.1 ([24, Proposition 5.1])**.** *Let* $\mathbf{A}$ *be a subdirectly irreducible CID-groupoid. Then there is a cancellation groupoid* $\mathbf{B}$ *such that either* $\mathbf{A} \cong \mathbf{B}$ *or* $\mathbf{A} \cong \mathbf{B}^{\infty}$*.*

In Theorem 4.1.1, $\mathbf{B}$ is a subalgebra of $\mathbf{A}$, so it is also a CID-groupoid. Also, if A is finite, then so is B. In the finite case $\mathbf{B}$, being cancellative, is a Latin square.

Let $xy^2 = (xy)y$ and inductively define $xy^{j+1} = (xy^j)y$. Let n be a positive integer, and define $\mathcal{V}_n$ to be the variety of all CID-groupoids satisfying the identity $xy^n \approx x$. Note that by taking $x/y = xy^{n-1}$ in $\mathcal{V}_n$ we have $(x/y) \cdot y \approx xy^n \approx x$. Combining this observation with commutativity we conclude that $\mathcal{V}_n$ is term-equivalent to a variety of quasigroups. In fact, $\mathcal{V}_2$ satisfies $(xy)y \approx x$, so it is the variety of *distributive* squags. From our discussion in Section 1.4, $\mathcal{V}_n$ is a strongly irregular, tractable variety. This sets the stage for a structure theorem for CID-groupoids.

Theorem 4.1.2. *Every finite CID-groupoid is a Płonka sum of Latin squares.*

Proof. Suppose that $\mathbf{A}$ is an arbitrary finite CID-groupoid. Let $m = |A|$ and set $n = m!$. Write $\mathbf{A}$ as a subdirect product of subdirectly irreducible algebras, $\mathbf{A}_i$, for $i \in I$. By Theorem 4.1.1, each $\mathbf{A}_i$ is isomorphic to either $\mathbf{B}_i$ or to $\mathbf{B}_i^\infty$, for some Latin square $\mathbf{B}_i$. Since $|B_i| \leq m$, it follows that $\mathbf{B}_i \in \mathcal{V}_n$. Consequently both $\mathbf{B}_i$ and $\mathbf{B}_i^\infty$ lie in $\widetilde{\mathcal{V}_n}$. Thus $\mathbf{A} \in \widetilde{\mathcal{V}_n}$, so by Theorem 2.1.3, $\mathbf{A}$ is a Płonka sum of Latin squares. □

Corollary 4.1.3. *The variety of CID-groupoids is tractable.*

Proof. By Theorem 4.1.2, every finite CID-groupoid lies in $\widetilde{\mathcal{V}_n}$ for some $n \in \omega$. By Corollary 2.2.2, $\widetilde{\mathcal{V}_n}$ is tractable. □

Corollary 4.1.4. *The variety of CIE-groupoids is tractable.*

Proof. Every idempotent, entropic groupoid is distributive, following from:

$$x(yz) \approx (xx)(yz) \approx (xy)(xz).$$

The result is then immediate following Corollary 4.1.3. □

4.2 Short Identities

Which other identities can serve as weakenings of associativity, so that tractability of the variety of CI-groupoids satisfying them may be determined? One possibility is to examine those groupoid identities $p \approx q$ such that

(i) the variables appearing in p and q are some subset of $\{x, y, z\}$

(ii) there are 3 or fewer variables appearing in p and q

(iii) no restriction is made to the ordering or grouping of the variables.

We will refer to these as *short* identities. In contrast to Bol-Moufang type identities, the variables need not appear in the same order on both sides of a short identity, and in fact a short identity may be irregular. We begin with a discussion identifying which terms and identities we need to consider, reduce the identities to five distinct equivalence classes, and investigate the tractability of each variety of CI-groupoids satisfying one additional such identity.

Since we are working in the context of CI-groupoids, a few simplifications may be made. Due to commutativity, we may assume that any three-variable term appearing in a short identity is right-associated (e.g. $x(yz)$), since it is equal to the corresponding left-associated term (e.g. $(yz)x$). Commutativity also allows us to consider only those terms where associated pairs of variables appear alphabetically. We may also assume that any associated pair of variables (in either a two- or three-variable term) is distinct, otherwise idempotence could be used to eliminate the pair. So, the only terms we need to consider are:

p_1	$x(xy)$	p_6	$y(yz)$	p_{11}	xz
p_2	$x(xz)$	p_7	$z(xy)$	p_{12}	yz
p_3	$x(yz)$	p_8	$z(xz)$	p_{13}	x
p_4	$y(xy)$	p_9	$z(yz)$	p_{14}	y
p_5	$y(xz)$	p_{10}	xy	p_{15}	z

We consider all possible identities $p_i \approx p_j$, $1 \leq i < j \leq 15$, excluding the obviously trivial $x \approx y$, $x \approx z$ and $y \approx z$. Using reassignment of variables, together with commutativity and idempotence, we are able to reduce the list of possibilities to just 22 identities. Lastly, as with Bol-Moufang groupoids, we check their equivalence using Prover9 and Mace4, and then analyze the complexity of the corresponding varieties. We use the following numbering convention for variable reassignments:

1. $x \leftrightarrow y$

2. $x \leftrightarrow z$

3. $y \leftrightarrow z$

4. $x \to y \to z \to x$

5. $x \to z \to y \to x$

For readability, we write each left-hand term a single time. When an identity is equivalent to one listed previously, we give the appropriate variable reassignment and equivalent identity.

$$x(xy) \approx \begin{cases} x(xz) \\ x(yz) \\ y(xy) \\ y(xz) \\ y(yz) \\ z(xy) \\ z(xz) \text{ is equivalent to } x(xy) \approx y(yz) \text{ by 4.} \\ z(yz) \\ xy \\ xz \\ yz \\ x \\ y \\ z \end{cases}$$

$$x(xz) \approx \begin{cases} x(yz) \text{ is equivalent to } x(xy) \approx x(yz) \text{ by 3.} \\ y(xy) \text{ is equivalent to } x(xy) \approx y(yz) \text{ by 1.} \\ y(xz) \text{ is equivalent to } x(xy) \approx z(xy) \text{ by 3.} \\ y(yz) \text{ is equivalent to } x(xy) \approx z(yz) \text{ by 3.} \\ z(xy) \text{ is equivalent to } x(xy) \approx y(xz) \text{ by 3.} \\ z(xz) \text{ is equivalent to } x(xy) \approx y(xy) \text{ by 3.} \\ z(yz) \text{ is equivalent to } x(xy) \approx y(yz) \text{ by 3.} \\ xy \text{ is equivalent to } x(xy) \approx xz \text{ by 3.} \\ xz \text{ is equivalent to } x(xy) \approx xy \text{ by 3.} \\ yz \text{ is equivalent to } x(xy) \approx yz \text{ by 3.} \\ x \text{ is equivalent to } x(xy) \approx x \text{ by 3.} \\ y \text{ is equivalent to } x(xy) \approx z \text{ by 3.} \\ z \text{ is equivalent to } x(xy) \approx y \text{ by 3.} \end{cases}$$

$$x(yz) \approx \begin{cases} y(xy) \text{ is equivalent to } x(xy) \approx y(xz) \text{ by 1.} \\ y(xz) \text{ is equivalent to } x(yz) \approx z(xy) \text{ by 3.} \\ y(yz) \text{ is equivalent to } x(xy) \approx z(xy) \text{ by 5.} \\ z(xy) \\ z(xz) \text{ is equivalent to } x(yz) \approx y(xy) \text{ by 3.} \\ z(yz) \text{ is equivalent to } x(yz) \approx y(yz) \text{ by 3.} \\ xy \\ xz \text{ is equivalent to } x(yz) \approx xy \text{ by 3.} \\ yz \\ x \\ y \\ z \end{cases}$$

$$y(xy) \approx \begin{cases} y(xz) \text{ is equivalent to } x(xy) \approx x(yz) \text{ by 1.} \\ y(yz) \text{ is equivalent to } x(xy) \approx x(xz) \text{ by 1.} \\ z(xy) \text{ is equivalent to } x(yz) \approx y(yz) \text{ by 2.} \\ z(xz) \text{ is equivalent to } x(xy) \approx z(yz) \text{ by 1.} \\ z(yz) \text{ is equivalent to } x(xy) \approx y(yz) \text{ by 2.} \\ xy \text{ is equivalent to } x(xy) \approx xy \text{ by 1.} \\ xz \text{ is equivalent to } x(xy) \approx yz \text{ by 1.} \\ yz \text{ is equivalent to } x(xy) \approx xz \text{ by 1.} \\ x \text{ is equivalent to } x(xy) \approx y \text{ by 1.} \\ y \text{ is equivalent to } x(xy) \approx x \text{ by 1.} \\ z \text{ is equivalent to } x(xy) \approx z \text{ by 1.} \end{cases}$$

$$y(xz) \approx \begin{cases} y(yz) \text{ is equivalent to } x(yz) \approx x(xy) \text{ by 5.} \\ z(xy) \text{ is equivalent to } x(yz) \approx z(xy) \text{ by 1.} \\ z(xz) \text{ is equivalent to } x(yz) \approx z(yz) \text{ by 1.} \\ z(yz) \text{ is equivalent to } x(yz) \approx z(xz) \text{ by 1.} \\ xy \text{ is equivalent to } x(yz) \approx xy \text{ by 1.} \\ xz \text{ is equivalent to } x(yz) \approx yz \text{ by 1.} \\ yz \text{ is equivalent to } x(yz) \approx xz \text{ by 1.} \\ x \text{ is equivalent to } x(yz) \approx y \text{ by 1.} \\ y \text{ is equivalent to } x(yz) \approx x \text{ by 1.} \\ z \text{ is equivalent to } x(yz) \approx z \text{ by 1.} \end{cases}$$

$$y(yz) \approx \begin{cases} z(xy) \text{ is equivalent to } x(yz) \approx y(xy) \text{ by 2.} \\ z(xz) \text{ is equivalent to } x(xz) \approx z(yz) \text{ by 1.} \\ z(yz) \text{ is equivalent to } x(xy) \approx y(xy) \text{ by 2.} \\ xy \text{ is equivalent to } x(xy) \approx xz \text{ by 5.} \\ xz \text{ is equivalent to } x(xy) \approx yz \text{ by 5.} \\ yz \text{ is equivalent to } x(xy) \approx xy \text{ by 5.} \\ x \text{ is equivalent to } x(xy) \approx z \text{ by 5.} \\ y \text{ is equivalent to } x(xy) \approx x \text{ by 5.} \\ z \text{ is equivalent to } x(xy) \approx y \text{ by 5.} \end{cases}$$

$$z(xy) \approx \begin{cases} z(xz) \text{ is equivalent to } x(yz) \approx x(xy) \text{ by 4.} \\ z(yz) \text{ is equivalent to } x(yz) \approx x(xy) \text{ by 2.} \\ xy \text{ is equivalent to } x(yz) \approx yz \text{ by 2.} \\ xz \text{ is equivalent to } x(yz) \approx xy \text{ by 4.} \\ yz \text{ is equivalent to } x(yz) \approx xy \text{ by 2.} \\ x \text{ is equivalent to } x(yz) \approx y \text{ by 4.} \\ y \text{ is equivalent to } x(yz) \approx y \text{ by 2.} \\ z \text{ is equivalent to } x(yz) \approx x \text{ by 2.} \end{cases}$$

$$z(xz) \approx \begin{cases} z(yz) \text{ is equivalent to } x(xy) \approx x(xz) \text{ by 2.} \\ xy \text{ is equivalent to } x(xy) \approx yz \text{ by 4.} \\ xz \text{ is equivalent to } x(xy) \approx xy \text{ by 4.} \\ yz \text{ is equivalent to } x(xy) \approx xz \text{ by 4.} \\ x \text{ is equivalent to } x(xy) \approx y \text{ by 4.} \\ y \text{ is equivalent to } x(xy) \approx z \text{ by 4.} \\ z \text{ is equivalent to } x(xy) \approx x \text{ by 4.} \end{cases}$$

$$z(yz) \approx \begin{cases} xy \text{ is equivalent to } x(xy) \approx yz \text{ by 2.} \\ xz \text{ is equivalent to } x(xy) \approx xz \text{ by 2.} \\ yz \text{ is equivalent to } x(xy) \approx xy \text{ by 2.} \\ x \text{ is equivalent to } x(xy) \approx z \text{ by 2.} \\ y \text{ is equivalent to } x(xy) \approx y \text{ by 2.} \\ z \text{ is equivalent to } x(xy) \approx x \text{ by 2.} \end{cases}$$

$$xy \approx \begin{cases} xz \\ yz \text{ is equivalent to } xy \approx xz \text{ by 1.} \\ x \\ y \text{ is equivalent to } xy \approx x \text{ by 1.} \\ z \end{cases}$$

$$xz \approx \begin{cases} yz \text{ is equivalent to } xy \approx xz \text{ by 2.} \\ x \text{ is equivalent to } xy \approx x \text{ by 3.} \\ y \text{ is equivalent to } xy \approx z \text{ by 3.} \\ z \text{ is equivalent to } xy \approx y \text{ by 3.} \end{cases}$$

$$yz \approx \begin{cases} x \text{ is equivalent to } xy \approx z \text{ by 2.} \\ y \text{ is equivalent to } xy \approx x \text{ by 5.} \\ z \text{ is equivalent to } xy \approx x \text{ by 2.} \end{cases}$$

We are left with just 22 short identities to check for equivalence:

$$x(xy) \approx \begin{cases} x(xz) \\ x(yz) \\ y(xy) \\ y(xz) \\ y(yz) \\ z(xy) \\ z(yz) \\ xy \\ xz \\ yz \\ x \\ y \\ z \end{cases}$$

$$x(yz) \approx \begin{cases} z(xy) \\ xy \\ yz \\ x \\ y \\ z \end{cases}$$

$$xy \approx \begin{cases} xz \\ x \\ z \end{cases}$$

Following the same procedure as in Chapter 3, we analyzed the equivalences between these identities, relative to the underlying variety of CI-groupoids. This yielded five equivalence classes.

Theorem 4.2.1. *The following short identities are equivalent, and determine the trivial variety of groupoids:* $x(xy) \approx x(xz)$, $x(xy) \approx x(yz)$, $x(xy) \approx y(xz)$, $x(xy) \approx y(yz)$, $x(xy) \approx z(xy)$, $x(xy) \approx xz$, $x(xy) \approx yz$, $x(xy) \approx x$, $x(xy) \approx z$, $x(yz) \approx xy$, $x(yz) \approx yz$, $x(yz) \approx x$, $x(yz) \approx y$, $x(yz) \approx z$, $xy \approx xz$, $xy \approx x$, $xy \approx z$.

Theorem 4.2.2. *The following short identities are equivalent, and determine the variety of squags:* $x(xy) \approx y$, $x(xy) \approx z(yz)$.

Proof. The identity $x(xy) \approx y$ is the squag law, examined earlier. Assuming this identity, we derive the second easily: $x(xy) \approx y \approx z(zy) \approx z(yz)$. Assuming the second identity holds, we can replace z by y, and apply idempotence to derive the squag law. □

Three identities remain, each distinct from the others.

Theorem 4.2.3. *The following short identities determine the varieties* $\mathcal{SL}$, $2\mathcal{SL}$, *and a variety we call* $\mathcal{S}_3$, *such that* $\mathcal{SL} \subset 2\mathcal{SL} \subset \mathcal{S}_3$: $x(yz) \approx z(xy)$, $x(xy) \approx xy$, *and* $x(xy) \approx y(xy)$.

Proof. The first identity is a commuted version of the associative law, while the second is the 2-semilattice law. We previously showed that $2\mathcal{SL}$ is strictly larger than $\mathcal{SL}$. We need to show that $x(xy) \approx xy \Rightarrow x(xy) \approx y(xy)$, but not the converse. Assuming the 2-semilattice law, the defining identity for $\mathcal{S}_3$ is easily derived: $x(xy) \approx xy \approx yx \approx y(yx) \approx y(xy)$. Figure 4.2(a) is a CI-groupoid which is in $\mathcal{S}_3 \setminus 2\mathcal{SL}$. The 2-semilattice law fails because $0(0 \cdot 1) \neq 0 \cdot 1$. □

We have thus shown that the only varieties of CI-groupoids defined by an additional short identity are the trivial variety, the variety $\mathcal{S}q$ of squags, $\mathcal{SL}$, $2\mathcal{SL}$, and $\mathcal{S}_3$. We will show that they are distinct at the end of the section. The tractability of the first four was shown in the previous chapter. Our name for the variety defined by $x(xy) \approx y(xy)$ is meant to suggest the following result.

Theorem 4.2.4. $\mathcal{S}_3$ *is congruence meet-semidistributive.*

Proof. As before, we identify a family of idempotent Maltsev conditions that is satisfied in $\mathcal{S}_3$, but is only true in the trivial variety of modules. The result then follows from Theorem 1.3.6.

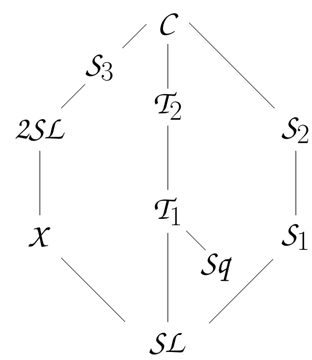

Figure 4.1 Tractable subvarieties of CI-groupoids

We take as our family the identities defining $\mathcal{S}_3$:

$$x \cdot x \approx x$$

$$x \cdot y \approx y \cdot x$$

$$x \cdot (x \cdot y) \approx y \cdot (x \cdot y).$$

Following from the discussion in Theorem 3.5.2, we again consider varieties modules over unital rings, such that the R-modules in question satisfy the implications $rx \approx 0 \Rightarrow r = 0_R$, $rx \approx x \Rightarrow r = 1_R$. Now, suppose that $\mathcal{M}$ is a variety of R-modules which has a binary term $x \cdot y$ satisfying the above Maltsev conditions. Such a term must be of the form $x \cdot y \approx rx + ry$. Interpreting the final axiom using this term yields the identity $r(x + r(x + y)) \approx r(y + r(x + y))$, or equivalently $r^2x - r^2y \approx 0$. Letting y be 0 gives $r^2x \approx 0$, which implies that $r^2 = 0_R$. However, $r + r = 1_R$, and squaring both sides yields

$$0_R = 4r^2 = (r + r)^2 = (1_R)^2 = 1_R,$$

so the ring R is trivial, and as a result the variety $\mathcal{M}$ must satisfy $x \approx 1_R x \approx 0_R x \approx 0$. That is, $\mathcal{M}$ must be trivial. □

The variety of squags is *not* congruence meet-semidistributive, and so after the previous theorem, we have justified that it is not contained in any of the other varieties of CI-groupoids defined by an additional short identity. The two-element semilattice is not a squag, so none of

	0	1	2	3
0	0	2	3	3
1		1	3	3
2			2	3
3				3

(a) $\mathcal{S}_3 \setminus \mathcal{2SL}$

	0	1	2
0	0	2	0
1		1	1
2			2

(b) $\mathcal{S}_1 \setminus \mathcal{S}_3$

	0	1	2
0	0	0	2
1		1	1
2			2

(c) $\mathcal{S}_3 \setminus \mathcal{S}_2$

Figure 4.2 Tables for Theorem 4.2.3 and Example 4.2.5

the reverse containments hold either. While $\mathcal{S}_3$ is SD($\wedge$), it is distinct from the congruence meet-semidistributive varieties of CI-groupoids of Bol-Moufang type investigated in the previous chapter.

Example 4.2.5. Figure 4.2(b) is a member of $\mathcal{S}_1$ which is not in $\mathcal{S}_3$. The $\mathcal{S}_3$ identity fails because $0(0\cdot 1) \neq 1\cdot(0\cdot 1)$. Figure 4.2(c) is a member of $\mathcal{S}_3$ which is not in $\mathcal{S}_2$. $B12$ $[x(y(xz)) \approx x((yx)z)]$ fails because $0(1(0\cdot 2)) \neq 0((1\cdot 0)2)$.

An updated version of Figure 3.2, including the nontrivial varieties of CI-groupoids determined by an additional short identity, is presented in Figure 4.1. As a consequence of Theorems 4.2.4 and 1.4.13, we have the following corollary, which settles the tractability of all varieties of CI-groupoids determined by an additional short identity.

Corollary 4.2.6. *$\mathcal{S}_3$ is tractable.*

4.3 CI-Groupoids of Generalized Bol-Moufang Type

Recall that a groupoid identity is of Bol-Moufang type if the same three variables appear on either side, one of the variables is repeated, the remaining two variables appear once, and the variables appear in the same order on either side. We drop the final condition as a further generalization. An identity $p \approx q$ is of *generalized Bol-Moufang type* if it satisfies the following:

(i) the same 3 variables appear in p and q,

(ii) one of the variables appears twice in p and q,

(iii) the remaining two variables appear once in p and q.

A variety of CI-groupoids is said to be of *generalized Bol-Moufang type* if it is defined by one additional identity which is of generalized Bol-Moufang type. In [12], the authors classify varieties of loops of generalized Bol-Moufang type, much in the same way that Phillips and Vojtěchovský classified the varieties of quasigroups and loops of Bol-Moufang type in [39] and [38]. In the present section we classify the varieties of CI-groupoids of generalized Bol-Moufang type which are *not* of Bol-Moufang type, with respect to the complexity of the corresponding CSP over algebras in each variety. The classification system for Bol-Moufang identities is easily extended to the generalized Bol-Moufang type. Since the variables might not appear in the same order on both sides of a generalized Bol-Moufang type identity, we can no longer assume that they are in alphabetical order. Accordingly, there are exactly 12 ways in which the 3 variables can form a word of length 4, and there are still only 5 ways in which a word of length 4 can be bracketed. They are:

A	$xxyz$	G	$xxzy$
B	$xyxz$	H	$xzxy$
C	$yxxz$	I	$zxxy$
D	$xyzx$	J	$xzyx$
E	$yxzx$	K	$zxyx$
F	$yzxx$	L	$zyxx$

1	$o(o(oo))$
2	$o((oo)o)$
3	$(oo)(oo)$
4	$(o(oo))o$
5	$((oo)o)o$

If $X, Y \in \{A, B, C, \dots, L\}$, and $i, j \in \{1, 2, 3, 4, 5\}$, let Xi be the groupoid term with variables ordered according to X and bracketed according to i, and let $XiYj$ be the identity $Xi \approx Yj$. For example, the identity $A1B2$ is $x(x(yz)) \approx x((yx)z)$. Notice that the variable orderings C, E and F differ from the classification system for identities of Bol-Moufang type in Section 3.1, but are equivalent under renaming the variables alphabetically in order of appearance. This ensures that the repeated variable is always x.

Some auxiliary terminology will be helpful in our classification. The variable order of term p is said to be *normal* if y appears before z (i.e. p is of variable order A—F). The remaining orders are called *flip*, because they are created from A—F by flipping y and z. We then name the identities $p \approx q$ as *normal-normal*, *normal-flip*, *flip-flip* and *flip-normal* depending on the orderings of p and q, respectively.

Table 4.1 Possible commutations

	1	2	3	4	5
A	D,F,G,J,L	B,D,E,H,J,K,L	F, G, L	B,C,I,K,L	C,I,L
B	D,E,F,G,H,J,K	A,D,E,H,J,K,L	C,D,E,H,I,J,K	A,C,I,K,L	A,C,I,K,L
C	E,F,G,H,I	F,G,I	B,D,E,H,I,J,K	A,I,L	A,B,I,K,L
D	B,E,F,G,H,J,K	A,F,G,J,L	B,C,E,H,I,J,K	A,F,G,J,L	A,B,E,H,J,K,L
E	C,F,G,H,I	C,F,G,H,I	B,C,D,H,I,J,K	B,D,F,G,H,J,K	A,B,D,H,J,K,L
F	C,G,I	C,E,G,H,I	A,G,L	B,D,E,G,H,J,K	A,D,G,J,L
G	A,D,F,J,L	B,D,E,F,H,J,K	A,F,L	C,E,F,H,I	C,F,I
H	A,B,D,E,J,K,L	B,D,E,F,G,J,K	B,C,D,E,I,J,K	C,E,F,G,I	C,E,F,G,I
I	A,B,C,K,L	A,C,L	B,C,D,E,H,J,K	C,F,G	C,E,F,G,H
J	A,B,D,E,H,K,L	A,D,F,G,L	B,C,D,E,H,I,K	A,D,F,G,L	B,D,E,F,G,H,K
K	A,B,C,I,L	A,B,C,I,L	B,C,D,E,H,I,J	A,B,D,E,H,J,L	B,D,E,F,G,H,J
L	A,C,I	A,B,C,I,K	A,F,G	A,B,D,E,H,J,K	A,D,F,G,J

As in Section 3.1, the *dual* $q' \approx p'$ of a groupoid identity $p \approx q$ is obtained by replacing occurrences of $\cdot$ with $\cdot^{op}$. For example, the dual of $A1B2$ is $(z(xy))x \approx ((zy)x)x$, the identity $K4L5$. One can easily identify the dual of any identity $XiYj$ of generalized Bol-Moufang type with the identity $Y'j'X'i'$ of Bol-Moufang type obtained by the rules:

$$A' = L, \quad B' = K, \quad C' = I, \quad D' = J, \quad E' = H, \quad F' = G$$

$$1' = 5, \quad 2' = 4, \quad 3' = 3.$$

Any commutative groupoid term is equal to its dual, and any groupoid identity is equivalent under commutativity to its dual. In the case of generalized Bol-Moufang type identities, every normal-normal identity is the dual of a flip-flip identity, and every normal-flip identity is the dual of a flip-normal identity. So we need only consider normal-normal and normal-flip identities. Every normal-flip identity is actually equivalent under commutativity to a normal-normal identity (since the dual of a term with flip variable order has normal variable order. So, in our quest to classify the varieties of CI-groupoids of generalized Bol-Moufang type, we only check normal-normal identities.

Those normal-normal identities $XiYj$ where $X = Y$ are of Bol-Moufang type. Letting X be a variable order and i a bracketing order, Table 4.1 lists in entry Xi all the possible variable orders which can result from commuting the variables in the term Xi. The same table is presented in [12], with an error in entry $C5$ that has been corrected here. Using this table, we can determine if a generalized Bol-Moufang identity $XiYj$ is equivalent to one of Bol-Moufang type by seeing if both Xi and Yj can be commuted to the *same* normal variable ordering.

Following this method, we found that only 24 of the normal-normal identities of generalized Bol-Moufang type are not immediately equivalent to one of Bol-Moufang type. They are: $A2C2$, $A2F1$, $A3B3$, $A3C3$, $A3D3$, $A3E3$, $A5B1$, $A5D1$, $A5E4$, $A5F4$, $B1C4$, $B2C2$, $B2F1$, $B3F3$, $C2D5$, $C2E5$, $C3F3$, $C4D1$, $C4E4$, $C4F4$, $D3F3$, $D5F1$, $E3F3$ and $E5F1$. They determine just *two* equivalence classes.

Theorem 4.3.1. *The following generalized Bol-Moufang identities are pairwise equivalent, and determine the variety of CID-groupoids:* $A3B3$*,* $A3C3$*,* $A3D3$*,* $A3E3$*,* $B3F3$*,* $C3F3$*,* $D3F3$ *and* $E3F3$*.*

Proof. In the presence of idempotence, the identity $A3B3$ $[(xx)(yz) \approx (xy)(xz)]$ is easily recognized as equivalent to the self-distributive law. The terms $B3$, $C3$, $D3$, and $E3$ are equal under commutativity, while $A3$ and $F3$ are equal as well. Each of the identities listed is obtained by swapping equal terms on one or both sides of the identity $A3B3$. □

Theorem 4.3.2. *The following generalized Bol-Moufang identities are pairwise equivalent, and determine a proper subvariety of CID-groupoids:* $A2C2$*,* $A2F1$*,* $A5B1$*,* $A5D1$*,* $A5E4$*,* $A5F4$*,* $B1C4$*,* $B2C2$*,* $B2F1$*,* $C2D5$*,* $C2E5$*,* $C4D1$*,* $C4E4$*,* $C4F4$*,* $D5F1$ *and* $E5F1$*.*

Proof. The terms $A2$, $B2$, $D5$, and $E5$ are equal under commutativity, and $C2$ is equal to $F1$ as well. $A5B1:$ $[((xx)y)z \approx x(y(xz))]$, which we rewrite using idempotence as $(xy)z \approx x(y(xz))$, implies distributivity (assuming commutativity and idempotence) as follows:

$$\begin{aligned} x(yz) &\approx (yz)x \\ &\approx y(z(yx)) \\ &\approx y((xy)z) \\ &\approx y(x(y(xz))) \\ &\approx (yx)(xz) \approx (xy)(xz) \end{aligned}$$

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

Figure 4.3 Table for Theorem 4.3.2

Thus, every CI-groupoid satisfying $A5B1$ (or any of the other equivalent identities listed) is distributive. The inclusion is proper. Figure 4.3 is a CID-groupoid which does not satisfy $A5B1$. It fails since $(0 \cdot 0)1 \neq 0(0(0 \cdot 1))$. □

Following from Theorems 4.3.1 and 4.3.2, we conclude that every variety of CI-groupoids of generalized Bol-Moufang type which is not of Bol-Moufang type is distributive. This gives our final result as an immediate consequence of Corollary 4.1.3.

Corollary 4.3.3. *Every variety of CI-groupoids of generalized Bol-Moufang type which is not of Bol-Moufang type is tractable.*

CHAPTER 5. FUTURE DIRECTIONS

In the preceding chapters, we provided a general preservation result for tractable constraint satisfaction problems over finite algebras, then applied it to several broad classes of identities weaker than the associative law. Each dimension of the present work suggests directions for further research, and in this chapter we examine them in turn. Some of the questions we pose will be easily resolved, while others will take considerable effort to investigate.

5.1 Other Varieties of Groupoids

Perhaps the most obvious next step is to further generalize the families of identities investigated in this thesis. The Bol-Moufang identities are a family of four-variable groupoid identities (where a single variable is repeated) of independent interest to quasigroup and loop theorists. In Section 4.3, we generalized the notion of Bol-Moufang identity by removing the requirement that the variable ordering be the same on both sides of an identity. Two other generalizations were investigated by REU students at Miami University [17]. First, we might be interested in classifying the varieties of CI-groupoids axiomatized by groupoid identities of the form $p(x, y, z, u) \approx q(x, y, z, u)$, where x, y, z and u are four distinct variables that each appear once, in the same order, in p and q. These identities are of the form Aij where $1 \leq i < j \leq 5$ indicate the way in which the left- and right-hand sides are parenthesized.

A	$xyzu$

1	$o(o(oo))$
2	$o((oo)o)$
3	$(oo)(oo)$
4	$(o(oo))o$
5	$((oo)o)o$

A second generalization investigated in [17] alternatively generalizes the notion of Bol-Moufang identity to those five-variable groupoid identities in which the same four variables appear on either side, one of the variables is repeated, the remaining three appear just once, and the variables appear in the same order on either side. This leads to 10 distinct variable orderings and 14 possible parenthesization patterns to consider.

A	$xxyzu$
B	$xyxzu$
C	$xyzxu$
D	$xyzux$
E	$xyyzu$
F	$xyzyu$
G	$xyzuy$
H	$xyzzu$
I	$xyzuz$
J	$xyzuu$

1	$o(o(o(oo)))$
2	$o(o((oo)o))$
3	$o((oo)(oo))$
4	$o((o(oo))o)$
5	$o(((oo)o)o)$
6	$(oo)(o(oo))$
7	$(oo)((oo)o)$
8	$((oo)o)(oo)$
9	$(o(oo))(oo)$
10	$(o(o(oo)))o$
11	$(o((oo)o))o$
12	$((oo)(oo))o$
13	$((o(oo))o)o$
14	$(((oo)o)o)o$

In the context of loops [17], many of these identities were shown to be stronger than the associative law, and so we speculate that a large number of them will define varieties of semi-lattices in the context of CI-groupoids. While the classification of the equivalences between identities of either of the above types may be of genuine interest to universal algebraists, it does not appear it will result in any interesting results for the constraint satisfaction problem over finite algebras from the varieties they define.

We might also wish to consider identities which are just slightly longer than the short identities we investigated in Section 4.2. One possibility would be to examine those groupoid identities $p \approx q$ such that

(i) the variables appearing in p and q are some subset of $\{x, y, z, u\}$

(ii) there are 4 or fewer variables appearing in p and q

(iii) no restriction is made to the ordering or grouping of the variables.

Many, but not all, of these identities are of the very first type outlined in the present section, and automated reasoning tools might be useful in their classification. There is, however, a limit to the utility of tools such as Prover9 in the analysis of these equivalences, as discussed in Appendix A. The proofs in Chapter 4 were produced without the aid of computers, with the exception of some counterexamples produced by Mace4. While a tool similar to Table 4.1 might aid in the reduction of five-variable generalized Bol-Moufang identities to a smaller number of cases, there are still 910 such identities. Even with computational assistance, any further generalization would greatly increase the of effort required to complete a classification, and the output produced might not be worth the time required to input the identities into Prover9.

A final way we might generalize the varieties of algebras studied in this thesis would be to remove one of the underlying assumptions (commutativity or idempotence). The classification of varieties of CI-groupoids of Bol-Moufang type was drastically simplified by the equivalence (under commutativity) of any Bol-Moufang type identity and its dual, and we speculate that the removal of commutativity would add a few new equivalence classes to our classification. As evidence of this, observe that the three identities defining $\mathcal{S}_2$ were shown to be equivalent under commutativity alone in Theorem 3.2.6.

Other proofs relied solely on the use of idempotence - Theorem 3.2.7 and Lemma 3.2.8, for example. In Section 4.1, we gave structural results for CID and CIE groupoids, relying on the fact that every idempotent, entropic groupoid is distributive. However, not every commutative, entropic groupoid is distributive, so the removal of idempotence in this case (as well as the others discussed in Chapter 4) might lead to additional structural results.

5.2 Structure of Congruence Meet-Semidistributive Varieties

Another approach to further research of a purely universal algebraic nature would be to investigate further those varieties of CI-groupoids of Bol-Moufang type which are SD($\wedge$). The theory of Płonka sums was applicable in the case of $\mathcal{T}_1$ (which is not SD(∧)), and generalized to provide some insight into the structure of $\mathcal{T}_2$. Our proofs that $2\mathcal{SL}$, $\mathcal{S}_2$, and $\mathcal{S}_3$ (and their subvarieties $\mathcal{X}$, $\mathcal{SL}$, and $\mathcal{S}_1$) are congruence meet-semidistributive relied on a result of Kearnes and Kiss which required us to produce certain classes of identities and show that they failed in any nontrivial variety of modules. To justify the tractability of a variety, however, we only require every *finite* algebra in the variety to be SD(∧). An algebra is said to be *locally finite* if every one of its finitely generated subalgebras is finite, and a variety is locally finite if every algebra therein is locally finite. Any variety which is generated by a finite set of finite algebras is locally finite, and so the following result of Kozik, Krokhin, Valeriote, and Willard suffices for a single finite algebra (and the variety it generates) to be SD(∧).

Theorem 5.2.1 ([26], Theorem 2.8)**.** *A locally finite variety is congruence meet-semidistributive if and only if it has* 3*-ary and* 4*-ary weak near-unanimity terms* $v(x, y, z)$ *and* $w(x, y, z, u)$ *that satisfy* $v(y, x, x) \approx w(y, x, x, x)$.

WNU terms for $\mathcal{S}_2$ satisfying the requirements of Theorem 5.2.1 are $v(x, y, z) = (xy)(z(xy))$ and $w(x, y, z, u) = (xy)(zu)$, and we obtain the tractability of $\mathcal{S}_2$ as a corollary of the previous theorem. Our use of the Kearnes and Kiss result allows a stronger structural conclusion about $\mathcal{S}_2$. It does, however, prompt the question: Which of the congruence meet-semidistributive varieties of CI-groupoids of Bol-Moufang type are locally finite? It is well known that the variety of semilattices is generated by the two-element semilattice (its only subdirectly irreducible member), so it is locally finite. We suspect that the free $\mathcal{S}_2$-algebra on two generators is infinite, which would imply that $\mathcal{S}_2$ is not locally finite, but do not yet have a proof. The variety of 2-semilattices is not locally finite, since the free 2-semilattice on three generators is clearly infinite (consider the sequence of elements x, xy, $(xy)z$, $((xy)z)x$, $(((xy)z)x)y, \dots$). A much more difficult question to settle would be whether or not the terms in Theorem 5.2.1 are sufficient to show that *any* variety (locally finite or otherwise) is congruence meet-semidistributive. This would give a characterization of congruence meet-semidistributivity more along the lines of those for congruence-permutable or arithmetical varieties.

In Theorem 3.6.7, we showed that $\mathcal{T}_1$ is the regularization of the variety of squags. Every

	0	1	2	3	4	5
0	0	2	1	3	5	4
1	2	1	0	3	5	4
2	1	0	2	4	3	5
3	3	3	4	3	5	4
4	5	5	3	5	4	3
5	4	4	5	4	3	5

Figure 5.1 Table for Example 5.2.3

variety of CI-groupoids of Bol-Moufang type is regular—that is, defined by a set of regular identities—so we can ask if any besides $\mathcal{T}_1$ is the regularization of some strongly irregular subvariety. One approach to this problem would be to examine candidates for the partition operation $x \vee y$ using Prover9 to check if they satisfy (P1)-(P5). For the meet-semidistributive varieties, the question is open. For the variety $\mathcal{T}_2$, the answer is known (in the negative), requiring the following result on subdirectly irreducible members of the regularization of a strongly irregular variety.

Theorem 5.2.2 ([27])**.** *Let $\mathcal{V}$ be a strongly irregular variety. The subdirectly irreducible members of $\widetilde{\mathcal{V}}$ are the algebras $\mathbf{A}$ and $\mathbf{A}^\infty$, as $\mathbf{A}$ ranges over all subdirectly irreducible algebras of $\mathcal{V}$, and the algebra 1^∞, where 1 denotes a trivial $\mathcal{V}$-algebra.*

Example 5.2.3. Figure 5.1 presents the smallest algebra in $\mathcal{T}_2 \setminus \mathcal{T}_1$. It is subdirectly irreducible, but not of the form required by Theorem 5.2.2. Thus, $\mathcal{T}_2$ is not the regularization of any strongly irregular variety.

5.3 CSP Results

Of course, no discussion of future research based upon the work in this thesis would be complete without paying special attention to Constraint Satisfaction Problems and the Algebraic Dichotomy Conjecture. As we mentioned in Section 1.5, proving Conjecture 1.5.4 would not only settle the original Feder and Vardi conjecture, but would also provide a characterization of all tractable algebras via a term condition. We describe our main result (Theorem 2.2.1) as preserving the tractability of CSPs by "pasting together" algebras form a tractable variety in a Płonka sum. This leads us to question if there are other, perhaps more general, preservation results for the tractability of CSPs.

One way of conceptualizing the regularization $\widetilde{\mathcal{V}}$ of a variety $\mathcal{V}$ is as the join $\mathcal{V} \vee \mathcal{Sl}_\rho$ in the lattice of varieties of type ρ, and we have shown that this method of construction preserves the tractability of a variety of algebras. The *Maltsev product* of two idempotent classes $\mathcal{A}$ and $\mathcal{B}$ of algebras of the same type is the class (not necessarily a variety)

$$\mathcal{A} \circ \mathcal{B} = \{\mathbf{A} \mid (\exists \theta \in \operatorname{Con} \mathbf{A})(\mathbf{A}/\theta \in \mathcal{B} \text{ and } (\forall a \in a)(a/\theta \in \mathcal{A}))\}.$$

The construction was introduced in [32]. If $\mathcal{Sq}$ is the variety of squags, our work in Section 3.6 can be interpreted as showing that $\mathcal{T}_1 = \mathcal{Sq} \vee \mathcal{SL}$ and $\mathcal{T}_2 \subseteq \mathcal{Sq} \circ \mathcal{SL}$, and both $\mathcal{Sq}$ and $\mathcal{SL}$ are tractable varieties. Many tractability results stemming from the algebraic approach to the CSP are based on the existence of terms satisfying particular Maltsev conditions. Freese and McKenzie have obtained some preliminary results regarding the preservation of some (but not all) of these term conditions under Maltsev product, and further investigation of such constructions may yield further preservation results. Some unpublished work of Maróti [34] shows the tractability of the many-sorted CSP where each "sort" is an algebra lying in the Maltsev product $\mathcal{A} \circ \mathcal{B}$ of an SD($\wedge$) variety $\mathcal{A}$ and a variety $\mathcal{B}$ possessing an edge term, lending further support to this line of inquiry.

Finally, we might consider CSP instances over algebras which have certain order-theoretic or graph-theoretic properties. In [6], Bulatov showed that the variety of 2-semilattices is tractable by first reducing to the case where a certain ordering imposed on the algebra produced a simple, strongly connected digraph. In the general case, Bulatov showed a similar reduction to that in our Theorem 2.2.1. Namely, that it is enough to search for satisfying assignments of the variables to values in the "greatest" connected component of the digraph structure imposed on the algebra. The result of [6] inspired key steps in the proof of the more general result of Barto and Kozik [2]. It is well known that the CSP Dichotomy Conjecture has an equivalent statement in terms of digraphs [14], so we speculate that the algebraic approach to CSP in conjunction with associated graph-theoretic properties will be a valuable source of future results.

APPENDIX A. AUTOMATED REASONING TOOLS

No paper which makes use of automated reasoning tools would be complete without a discussion of their place within serious mathematical research. Issues of interpretation, presentation, and ease of use should be considered when choosing to implement such tools. In this appendix, we discuss these and other issues, and end with a detailed explanation of how the proofs of some results in Section 3.6 were translated from the raw output of Prover9 [36] to a more readable form. Many of these derivations are presented in Appendix B.

Michael Kinyon wrote in his research statement that "...the point of mathematics is to improve human understanding, and such understanding comes not just from the statements of theorems, but from knowing their proofs." Why, then should we pursue computer-aided mathematics? Kinyon points out that certain areas of algebra are young enough, and have unsolved problems which may be stated in purely equational form, and hence are amenable to computer attacks. Computer-aided mathematics can be viewed as a dialogue between the mathematician and the computer. One might obtain a few insights into, say, the theory of CI-groupoids as we have in the preceding chapters, use them to inform the input into an automated reasoning tool or model builder, and further interpret the output using human insight before giving it back to the computer as new input. This view of automated reasoning tools as a "lab assistant" was also expressed by Hart and Kunen in the introduction to [16]. In our case, after obtaining a classification of the varieties of CI-groupoids of Bol-Moufang type, we examined page after page of models (generated by Mace4) in the varieties $\mathcal{T}_1$ and $\mathcal{T}_2$ before seeing the importance of Płonka sums. Once we conjectured a choice for the partition operation in $\mathcal{T}_1$, Prover9 made short work of verifying that it satisfies (P1)–(P5).

A successful use of automated reasoning was performed by Stanovský [46], who used Prover9 to obtain a purely equational (alternate) proof of a result about distributive groupoids that previously required great structural insight to reduce the proof to a special case. However, as Stanovský noted in Section 3, of [46], despite all the our efforts, it might not be reasonable to simplify a machine-generated proof. We might also reach a point of diminishing returns, where additional effort to simplify the proof will provide no greater understanding of the objects at hand, or the proof itself is so long that even a humanized form would not be instructive. What is the researcher's best option in this case? Computer-generated proofs are, as the authors point out in [39], "cumbersome and difficult to read," so they are often relegated (in unedited form) to appendices and authors' web pages for only the most curious reader to access and interpret. Even if we are able to split a complicated proof into many shorter lemmas (our own Theorem 3.6.3 required 13), is a simplified proof worth presenting if the reasoning itself doesn't add to our understanding of the structures involved? Do they need to add to our understanding, or is it enough to simply produce proofs in a form more easily parsed by the nonspecialist? We found it much more satisfying to produce fully humanized proofs of all of our results which were first aided by Prover9, and were successful in this project.

While the process of humanization itself took several days of work, we feel it was simplified by a few particulars of the problem at hand. First and foremost, although more powerful automated reasoning tools are available, Prover9 and Mace4 are widely known for their ease of use. Their simple input language allowed for the quick modification of assumptions and goals which was necessary in order to rapidly identify the equivalent varieties of CI-groupoids of Bol-Moufang type. Once the equivalences were determined, it was a simple process to verify the equivalences by hand. Second, the fact that we investigated algebras involving a single binary operation further simplified the input and verification process. Prover9 and its predecessor (Otter) provide a host of additional features, including the option to include user-created "hints" for the software to use in its search for a proof. Adding even a single additional operation greatly increases the complexity of both input and output (evident from the relatively few humanized proofs in [39]), and also seemingly requires the use of the aforementioned additional features (evident in [16]), something we were able to avoid. Finally, Theorem 3.6.7 required us to prove that five distinct identities hold in $\mathcal{T}_1$. By recognizing which statements were repeatedly utilized in the computer-generated proof of these identities, we identified several of the intermediate identities in Lemma 3.6.1, focusing our humanization efforts first on unpacking those key lemmas before approaching the main result. We conclude this appendix with some specific comments about the humanization process for the proofs in Appendix B.

We illustrate the humanization process through a single example from the proof of Theorem 3.6.7. In order to show that the term $x \vee y = y(xy)$ satisfies (P5) $[(xy) \vee z \approx (x \vee z)(y \vee z)]$ in $\mathcal{T}_1$, we input into Prover9 the assumptions

```
x * x = x              # label(idem).
x * y = y * x          # label(comm).
x * (x * (y * z)) = (x * (x * y)) * z        # label(A14).
```

and the single goal

```
z*((x*y)*z) = (z*(x*z)) * (z*(y*z)) #label(Plonka5).
```

Prover9 produced the following output (with all settings left in their default).

```
1 x * ((y * z) * x) = (x * (y * x)) * (x * (z * x)) # label(Plonka5)
# label(non_clause) # label(goal).  [goal].
2 x * x = x # label(idem).  [assumption].
3 x * y = y * x # label(comm).  [assumption].
4 x * (x * (y * z)) = (x * (x * y)) * z # label(A14).  [assumption].
5 (x * (x * y)) * z = x * (x * (y * z)).  [copy(4),flip(a)].
6 (c1 * (c2 * c1)) * (c1 * (c3 * c1)) != c1 * ((c2 * c3) * c1)
# label(Plonka5) # answer(Plonka5).  [deny(1)].
7 c1 * (c1 * (c2 * (c1 * (c1 * c3)))) != c1 * (c1 * (c2 * c3))
# answer(Plonka5).  [copy(6),rewrite([3(4),3(9),5(11),3(17)])].
9 x * (x * (x * y)) = x * y.
[para(2(a,1),5(a,1,1,2)),rewrite([2(1)]),flip(a)].
12 x * (y * (y * z)) = y * (y * (z * x)).
[para(5(a,1),3(a,1)),flip(a)].
51 $F # answer(Plonka5).
```

```
[para(12(a,1),7(a,1,2,2)),rewrite([3(7),9(10)]),xx(a)].
```

Lines 1 through 4 are clearly labeled as the goal and our three assumptions. Line 5 gives as its justification `[copy(4),flip(a)]`, indicating that it is a *copy* of Line 4, while the `flip(a)` indicates that the line just copied was flipped about its first (ath) relation symbol (the equals sign). Prover9 works "from the outside in," attempting to produce a contradiction, and so Line 6 is just the assumption (for contradiction) that `c1`, `c2`, and `c3` are constants which violate Line 1. Line 7 is justified by `[copy(6),rewrite([3(4),3(9),5(11),3(17)])]`, indicating that Line 6 was first copied, and then rewritten by using Line 3 twice, then Line 5, then Line 3. The parentheticals (e.g., the (4) in 3(4)) are merely an internal Prover9 reference. The uses of Line 3 to commute the constants are obvious, and in the use of Line 5, the software has identified z with `c1*(c1*c3))`, x with `c1`, and y with `c2`. The unwound steps leading to Line 7 are:

```
(c1 * (c2 * c1)) * (c1 * (c3 * c1)) != c1 * ((c2 * c3) * c1)
(c1 * (c1 * c2)) * (c1 * (c3 * c1)) != c1 * ((c2 * c3) * c1)
(c1 * (c1 * c2)) * (c1 * (c1 * c3)) != c1 * ((c2 * c3) * c1)
c1 * (c1 * (c2 * (c1 * (c1 * c3)))) != c1 * ((c2 * c3) * c1)
c1 * (c1 * (c2 * (c1 * (c1 * c3)))) != c1 * (c1 * (c2 * c3))
```

Paramodulation (denoted by `para` in the output) is the key step repeatedly performed in a Prover9 proof. It is, roughly speaking, an inference rule which combines variable instantiation and substitution of equalities into a single step. Often such inferences are the most difficult steps of the proof to interpret, and Prover9 offers the option to expand a proof by filling in the multiple steps taken in a single paramodulation. Sometimes, though, this offers no greater insight. For Line 9, the instruction `[para(2(a,1),5(a,1,1,2)),rewrite([2(1)]),flip(a)]` can be broken down as:

- `para(2(a,1),5(a,1,1,2))` indicates instantiation of Line 5, then identification (in order to apply Line 2) of the right-hand factor (2) of the left-hand factor (1) of the left-hand side (1) of the first (ath relation symbol (otherwise identified as `x*x`) with the left-hand (1) side of Line 2.

- `rewrite([2(1)])` indicates to further rewrite using Line 2.

- `flip(a)` indicates to simply flip the statement about the equals sign.

In this case, the instantiation is clear. In Line 5, we replace y with x and z with y, obtaining the following unwound steps leading to Line 9:

```
(x * (x * x)) * y = x * (x * (x * y))
(x * x) * y = x * (x * (x * y))
x * y = x * (x * (x * y))
x * (x * (x * y)) = x * y
```

To obtain Line 12, we follow `[para(5(a,1),3(a,1)),flip(a)]`. That is, instantiate Line 3, identify its left-hand side with the left-hand side of Line 5 (then apply it), and to flip the resulting statement. The successive steps are thus:

```
(y * (y * z)) * x = x * (y * (y * z))
y * (y * (z * x)) = x * (y * (y * z))
x * (y * (y * z)) = y * (y * (z * x))
```

The final step (Line 51) in the computer-generated proof indicates that Prover9 has reached a contradiction, with the justification for this line indicating how to derive the contradictory statement. In this case, `[para(12(a,1),7(a,1,2,2)),rewrite([3(7),9(10)]),xx(a)]` indicates that we should instantiate Line 7, identify the right-hand factor of the right-hand factor of its left-hand side with the left-hand side of 12, and apply it. Then, rewrite the result further using Lines 3 and 9, to ultimately arrive at a contradiction. The final unwinding:

```
c1 * (c1 * (c2 * (c1 * (c1 * c3)))) != c1 * (c1 * (c2 * c3))
c1 * (c1 * (c1 * (c1 * (c3 * c2)))) != c1 * (c1 * (c2 * c3))
c1 * (c1 * (c1 * (c1 * (c2 * c3)))) != c1 * (c1 * (c2 * c3))
c1 * (c1 * (c2 * c3)) != c1 * (c1 * (c2 * c3))
```

The last line asserts that `c1 * (c1 * (c2 * c3))` is not equal to itself (which is clearly false), so Prover9 concludes that the goal follows from the assumptions. However, we have just begun our humanization. After understanding just what Prover9 has done, we must reconstruct a derivation of (P5) from the inside out. Line 51 can alternately be interpreted as a proof that the identity $z(z(x(z(zy)))) \approx z(z(xy))$, where the right hand side is clearly a commutation of the term $(xy) \vee z$. The proof consists of the string of equalities

$$z(z(x(z(zy)))) \overset{12}{\approx} z(z(z(z(yx)))) \overset{C}{\approx} z(z(z(z(xy)))) \overset{9}{\approx} z(z(xy)).$$

We can continue to unwind by following the above analysis to prove the specific instances of Line 12 and Line 9 required. They are $x(z(zy)) \overset{C}{\approx} (z(zy))x \overset{5}{\approx} z(z(yx))$ and $z(z(z(z(xy)))) \overset{5}{\approx} z((z(zz))(xy)) \overset{I}{\approx} z(z(xy))$ respectively. All that remains is to prove the identity

$$z(z(x(z(zy)))) \approx (z(xz))(z(yz)) \approx (x \vee z)(y \vee z).$$

But this is just a combination of Lines 6 and 7, interpreted as identities instead of negations. Following the same reconstruction procedure we have just outlined, and piecing together the particular instances we have justified above results in the following derivation of (P5), which we first displayed at the end of Chapter 3.

$$\begin{aligned}(xy) \vee z \approx z((xy)z) \approx z(z(yx)) &\approx z((z(zz))(yx)) \\ &\overset{A14}{\approx} z(z(z(z(yx)))) \overset{A14}{\approx} z(z((z(zy))x)) \approx z(z(x(z(zy)))) \\ &\overset{A14}{\approx} (z(zx))(z(zy)) \approx (z(xz))(z(yz)) \approx (x \vee z)(y \vee z).\end{aligned}$$

APPENDIX B. PROOFS

We present the equational derivations justifying some of the results from Section 3.6.

Proof of Lemma 3.6.1.

$$
\begin{aligned}
x(y(yx)) &\approx (y(yx))x \\
&\approx (((yy)y)(yx))x \\
&\overset{C15}{\approx} (y(y(y(yx))))x \\
&\approx (((y(yx))y)y)x \\
&\overset{C15}{\approx} (y(yx))(y(yx)) \\
&\approx y(yx)
\end{aligned}
$$

$$
\begin{aligned}
x(y(x(x(y(x(xz)))))) &\overset{C15}{\approx} x(((yx)x)(y(x(xz)))) \\
&\overset{C15}{\approx} x(((yx)x)(((yx)x)z)) \\
&\overset{C15}{\approx} ((x((yx)x))((yx)x))z \\
&\approx ((x(x(xy)))((yx)x))z \\
&\overset{C15}{\approx} ((((xx)x)y)((yx)x))z \\
&\approx ((xy)((yx)x))z \\
&\approx (((yx)x)(xy))z \\
&\overset{C15}{\approx} (y(x(x(xy))))z \\
&\overset{C15}{\approx} (y(((xx)x)y))z \\
&\approx (y(xy))z \\
&\approx ((xy)y)z
\end{aligned}
$$

$$\stackrel{C15}{\approx} x(y(yz))$$

$$\begin{aligned}
x(y(yz)) &\stackrel{C15}{\approx} ((xy)y)z \\
&\approx (y(xy))z \\
&\approx [[(y(xy))(y(xy))](y(xy))]z \\
&\stackrel{C15}{\approx} (y(xy))[(y(xy))[(y(xy))z]] \\
&\approx ((xy)y)[((xy)y)[((xy)y)z]] \\
&\stackrel{C15}{\approx} x[y[y[((xy)y)[((xy)y)z]]]] \\
&\stackrel{C15}{\approx} x[y[[(y((xy)y))((xy)y)]z]] \\
&\approx x[y[[(y(y(yx)))((xy)y)]z]] \\
&\stackrel{C15}{\approx} x[y[[(((yy)y)x)((xy)y)]z]] \\
&\approx x[y[[(yx)((xy)y)]z]] \\
&\approx x[y[[((xy)y)(yx)]z]] \\
&\stackrel{C15}{\approx} x[y[[x(y(y(yx)))]z]] \\
&\stackrel{C15}{\approx} x[y[[x(((yy)y)x)]z]] \\
&\approx x[y[[x(yx)]z]] \\
&\approx x[y[[(yx)x]z]] \\
&\stackrel{C15}{\approx} x(y(y(x(xz))))
\end{aligned}$$

$$\begin{aligned}
(xy)(x(xz)) &\stackrel{C15}{\approx} (((xy)x)x)z \\
&\approx (x(x(xy)))z \\
&\stackrel{C15}{\approx} (((xx)x)y)z \\
&\approx (xy)z
\end{aligned}$$

$$\begin{aligned}
x[y(y(z(zu)))] &\stackrel{(3.2)}{\approx} x[y(y(z(y(y(z(y(yu)))))))] \\
&\stackrel{C15}{\approx} x[y(y(((zy)y)(z(y(yu)))))] \\
&\stackrel{C15}{\approx} x[y(y(((zy)y)(((zy)y)u)))]
\end{aligned}$$

$$
\begin{aligned}
&\overset{C15}{\approx} x[y(((y((zy)y))((zy)y))u)] \\
&\approx x[y(((y(yz))(y(y(yz))))u)] \\
&\overset{C15}{\approx} x[y(((y(yz))(((yy)y)z))u)] \\
&\approx x[y(((y(yz))(yz))u)] \\
&\overset{C15}{\approx} x[y(y((yz)((yz)u)))] \\
&\overset{C15}{\approx} ((xy)y)[(yz)((yz)u)] \\
&\overset{C15}{\approx} [(((xy)y)(yz))(yz)]u \\
&\overset{C15}{\approx} [(x(y(y(yz))))(yz)]u \\
&\overset{C15}{\approx} [(x(((yy)y)z))(yz)]u \\
&\approx [(x(yz))(yz)]u \\
&\overset{C15}{\approx} x[(yz)((yz)u)] \\
&\approx x[(yz)(u(yz))]
\end{aligned}
$$

$$
\begin{aligned}
x(y(z(z(y(z(zu)))))) &\overset{C15}{\approx} x(((yz)z)(y(z(zu)))) \\
&\overset{C15}{\approx} x(((yz)z)(((yz)z)u)) \\
&\overset{C15}{\approx} ((x((yz)z))((yz)z))u \\
&\approx (((yz)z)(x(z(zy))))u \\
&\overset{(3.1)}{\approx} (((yz)z)(x(y(z(zy)))))u \\
&\overset{(3.1)}{\approx} (((yz)z)(x(y(y(z(zy))))))u \\
&\approx ((x(y(y(z(zy)))))(z(zy)))u \\
&\overset{C15}{\approx} ((((xy)y)(z(zy)))(z(zy)))u \\
&\overset{C15}{\approx} ((xy)y)((z(zy))((z(zy))u)) \\
&\overset{C15}{\approx} x(y(y((z(zy))((z(zy))u)))) \\
&\overset{C15}{\approx} x(y(((y(z(zy)))(z(zy)))u)) \\
&\overset{(3.1)}{\approx} x(y(((z(zy))(z(zy)))u)) \\
&\approx x(y((z(zy))u))
\end{aligned}
$$

$$
\begin{aligned}
&\approx x(y(((yz)z)u)) \\
&\overset{C15}{\approx} x(y(y(z(zu))))
\end{aligned}
$$

$$
\begin{aligned}
x(y(x(z(zy)))) &\approx (y(x(z(zy))))x \\
&\overset{(3.1)}{\approx} (y(z(y(z(zy)))))x \\
&\overset{(3.1)}{\approx} (y(x(y(y(z(zy))))))x \\
&\overset{C15}{\approx} (y(((xy)y)(z(zy))))x \\
&\approx (y((y(yx))(z(zy))))x \\
&\overset{C15}{\approx} (y((((y(yx))z)z)y))x \\
&\approx (((z((y(yx))z))y)y)x \\
&\overset{C15}{\approx} (z((y(yx))z))(y(yx)) \\
&\approx (((y(yx))z)z)(y(yx)) \\
&\overset{C15}{\approx} (y(yx))(z(z(y(yx)))) \\
&\overset{(3.1)}{\approx} z(z(y(yx)))
\end{aligned}
$$

$$
\begin{aligned}
x(y(y(z(y(yx))))) &\overset{C15}{\approx} ((xy)y)(z(y(yx))) \\
&\approx (y(yx))((y(yx))z) \\
&\approx ((y(yx))(y(yx)))((y(yx))z) \\
&\overset{(3.1)}{\approx} ((x(y(yx)))(y(yx)))((y(yx))z) \\
&\overset{C15}{\approx} x((y(yx))((y(yx))((y(yx))z))) \\
&\overset{C15}{\approx} x((((y(yx))(y(yx)))(y(yx)))z) \\
&\approx x((y(yx))z) \\
&\approx x(z(y(yx)))
\end{aligned}
$$

$$
\begin{aligned}
(x(y(yz)))(y(yu)) &\overset{(3.4)}{\approx} (x(y(yz)))(x(x(y(yu)))) \\
&\overset{(3.6)}{\approx} (x(y(yz)))(x(y(y(x(y(yu)))))) \\
&\overset{C15}{\approx} (x(y(yz)))(((xy)y)(x(y(yu))))
\end{aligned}
$$

$$\begin{aligned}
&\stackrel{C15}{\approx} (x(y(yz)))(((xy)y)(((xy)y)u)) \\
&\stackrel{C15}{\approx} (((xy)y)z)(((xy)y)(((xy)y)u)) \\
&\stackrel{(3.4)}{\approx} (((xy)y)z)u \\
&\stackrel{C15}{\approx} (x(y(yz)))u
\end{aligned}$$

$$\begin{aligned}
x(y(y(z(zx)))) &\approx x(y(((yy)y)(z(zx)))) \\
&\stackrel{C15}{\approx} x(y(y(y(y(z(zx)))))) \\
&\stackrel{(3.5)}{\approx} x(y(y((yz)(x(yz))))) \\
&\stackrel{C15}{\approx} ((xy)y)((yz)(x(yz))) \\
&\approx ((xy)y)((yz)(x(((yy)y)z))) \\
&\stackrel{C15}{\approx} ((xy)y)((yz)(x(y(y(yz))))) \\
&\stackrel{C15}{\approx} ((xy)y)((yz)(((xy)y)(yz))) \\
&\approx (y(xy))((yz)((yz)(y(xy)))) \\
&\stackrel{(3.1)}{\approx} (yz)((yz)(y(xy))) \\
&\approx (yz)(((xy)y)(yz)) \\
&\stackrel{C15}{\approx} (yz)(x(y(y(yz)))) \\
&\stackrel{C15}{\approx} (yz)(x(((yy)y)z)) \\
&\approx (yz)(x(yz)) \\
&\approx (yz)(x(y(z(zz)))) \\
&\stackrel{C15}{\approx} (yz)(x(((yz)z)z)) \\
&\approx (yz)((((yz)z)z)x) \\
&\stackrel{C15}{\approx} (yz)((yz)(z(zx))) \\
&\stackrel{(3.1)}{\approx} (z(zx))((yz)((yz)(z(zx)))) \\
&\stackrel{C15}{\approx} (z(zx))((yz)((((yz)z)z)x)) \\
&\stackrel{C15}{\approx} (z(zx))((yz)((y(z(zz)))x)) \\
&\approx (z(zx))((yz)((yz)x))
\end{aligned}$$

$$
\begin{aligned}
&\stackrel{C15}{\approx} (((z(zx))(yz))(yz))x \\
&\approx (((z(zx))(((yy)y)z))(yz))x \\
&\stackrel{C15}{\approx} (((z(zx))(y(y(yz))))(yz))x \\
&\stackrel{C15}{\approx} (((((z(zx))y)y)(yz))(yz))x \\
&\stackrel{C15}{\approx} (((z(zx))y)y)((yz)((yz)x)) \\
&\stackrel{C15}{\approx} (z(zx))(y(y((yz)((yz)x)))) \\
&\stackrel{C15}{\approx} (z(zx))(y(((y(yz))(yz))x)) \\
&\approx (z(zx))(y(((y(yz))(((yy)y)z))x)) \\
&\stackrel{C15}{\approx} (z(zx))(y(((y(yz))(y(y(yz))))x)) \\
&\approx (z(zx))(y(((y(y(yz)))(y(yz)))x)) \\
&\stackrel{C15}{\approx} (z(zx))(y(y((y(yz))((y(yz))x)))) \\
&\approx (z(zx))(y(y(((zy)y)(((zy)y)x)))) \\
&\stackrel{C15}{\approx} (z(zx))(y(y(((zy)y)(z(y(yx)))))) \\
&\stackrel{C15}{\approx} (z(zx))(y(y(z(y(y(z(y(yx)))))))) \\
&\stackrel{(3.2)}{\approx} (z(zx))(y(y(z(zx)))) \\
&\stackrel{(3.1)}{\approx} y(y(z(zx)))
\end{aligned}
$$

$$
\begin{aligned}
(xy)(z(xy)) &\approx (xy)(z(x(y(yy)))) \\
&\stackrel{C15}{\approx} (xy)(z(((xy)y)y)) \\
&\approx (xy)(z(y(y(xy)))) \\
&\stackrel{C15}{\approx} (xy)(((zy)y)(xy)) \\
&\approx (xy)((xy)(y(zy))) \\
&\stackrel{(3.1)}{\approx} (y(zy))((xy)((xy)(y(zy)))) \\
&\approx (y(zy))((xy)((xy)(y(yz)))) \\
&\stackrel{C15}{\approx} (y(zy))((xy)((((xy)y)y)z)) \\
&\stackrel{C15}{\approx} (y(zy))((xy)((x(y(yy)))z))
\end{aligned}
$$

$$
\begin{aligned}
&\approx (y(zy))((xy)((xy)z)) \\
&\overset{C15}{\approx} (((y(zy))(xy))(xy))z \\
&\approx ((xy)((y(zy))(xy)))z \\
&\approx ((xy)((y(zy))(((xx)x)y)))z \\
&\overset{C15}{\approx} ((xy)((y(zy))(x(x(xy)))))z \\
&\overset{C15}{\approx} ((xy)((((y(zy))x)x)(xy)))z \\
&\approx (((x((y(zy))x))(xy))(xy))z \\
&\overset{C15}{\approx} (x((y(zy))x))((xy)((xy)z)) \\
&\approx (((y(zy))x)x)((xy)((xy)z)) \\
&\overset{C15}{\approx} (y(zy))(x(x((xy)((xy)z)))) \\
&\overset{C15}{\approx} (y(zy))(x(((x(xy))(xy))z)) \\
&\approx (y(zy))(x(((x(xy))(((xx)x)y))z)) \\
&\overset{C15}{\approx} (y(zy))(x(((x(xy))(x(x(xy))))z)) \\
&\approx (y(zy))(x(((x(x(xy)))(x(xy)))z)) \\
&\overset{C15}{\approx} (y(zy))(x(x((x(xy))((x(xy))z)))) \\
&\approx (y(zy))(x(x(((yx)x)(((yx)x)z)))) \\
&\overset{C15}{\approx} (y(zy))(x(x(((yx)x)(y(x(xz)))))) \\
&\overset{C15}{\approx} (y(zy))(x(x(y(x(x(y(x(xz)))))))) \\
&\overset{(3.2)}{\approx} (y(zy))(x(x(y(yz)))) \\
&\approx ((zy)y)(x(x(y(yz)))) \\
&\overset{C15}{\approx} z(y(y(x(x(y(yz)))))) \\
&\overset{(3.3)}{\approx} z(y(y(x(xz))))
\end{aligned}
$$

$$
\begin{aligned}
x(x(y(yz))) &\overset{(3.11)}{\approx} (yx)(z(yx)) \\
&\approx (xy)(z(xy)) \\
&\overset{(3.11)}{\approx} y(y(x(xz)))
\end{aligned}
$$

□

Proof of Lemma 3.6.2.

$$
\begin{aligned}
x(x(y(yz))) &\overset{(3.1)}{\approx} (y(yz))(x(x(y(yz)))) \\
&\overset{(3.10)}{\approx} (y(yz))(z(x(x(y(yz))))) \\
&\overset{C15}{\approx} (y(yz))(((zx)x)(y(yz))) \\
&\approx (y(yz))((y(yz))(x(xz))) \\
&\overset{(3.11)}{\approx} (x(y(yz)))(z(x(y(yz)))) \\
&\overset{(3.9)}{\approx} (x(y(yz)))(y(y(z(x(y(yz)))))) \\
&\overset{C15}{\approx} (((x(y(yz)))y)y)(z(x(y(yz)))) \\
&\approx (y(y(x(y(yz)))))(z(x(y(yz)))) \\
&\overset{(3.4)}{\approx} (y(y(x(y(yz)))))(y(y(z(x(y(yz)))))) \\
&\overset{(3.8)}{\approx} (y(y(x(y(yz)))))(y(y(z(y(y(x(y(yz)))))))) \\
&\overset{C15}{\approx} (y(y(x(y(yz)))))(y(y(((zy)y)(x(y(yz)))))) \\
&\approx (y(y(x(y(yz)))))(y(y((x(y(yz)))(y(yz))))) \\
&\overset{(3.4)}{\approx} (y(y(x(y(yz)))))(y(y((x(y(yz)))(x(x(y(yz))))))) \\
&\approx (y(y(x(y(yz)))))(y(y((x(y(yz)))((x(y(yz)))x)))) \\
&\overset{(3.5)}{\approx} (y(y(x(y(yz)))))((y(x(y(yz))))(x(y(x(y(yz)))))) \\
&\approx (y(y(x(y(yz)))))((y(x(y(yz))))((y(x(y(yz))))x)) \\
&\overset{C15}{\approx} (((y(y(x(y(yz)))))(y(x(y(yz)))))(y(x(y(yz)))))x \\
&\overset{C15}{\approx} (y((y(x(y(yz))))((y(x(y(yz))))(y(x(y(yz)))))))x \\
&\approx (y(y(x(y(yz)))))x \\
&\approx x(y(y(x(y(yz))))) \\
&\overset{C15}{\approx} ((xy)y)(x(y(yz))) \\
&\overset{C15}{\approx} ((xy)y)(((xy)y)z) \\
&\approx (y(xy))(z(y(xy)))
\end{aligned}
$$

□

Proof of Theorem 3.6.3. We need to show that $x \vee y = y(xy)$ satisfies identities (P1)–(P4) in

Theorem 2.1.3. In order, they are:

$$(\text{P1})\ x \vee x \approx x:$$

$$\begin{aligned} x \vee x &\approx x(xx) \\ &\approx x \end{aligned}$$

$$(\text{P2})\ x \vee (y \vee z) \approx (x \vee y) \vee z:$$

$$\begin{aligned}
x \vee (y \vee z) &\approx x \vee (z(yz)) \\
&\approx (z(yz))(x(z(yz))) \\
&\approx (z(zy))(x(z(zy))) \\
&\approx ((z(zy))(x(z(zy))))((z(zy))(x(z(zy)))) \\
&\approx ((x(z(zy)))(z(zy)))((z(zy))(x(z(zy)))) \\
&\overset{C15}{\approx} x((z(zy))((z(zy))((z(zy))(x(z(zy)))))) \\
&\approx x((z(zy))((z(zy))((z(zy))((z(zy))x)))) \\
&\overset{C15}{\approx} x((z(zy))((((z(zy))(z(zy)))(z(zy)))x)) \\
&\approx x((z(zy))((z(zy))x)) \\
&\approx x((x(z(zy)))(z(zy))) \\
&\overset{C15}{\approx} x((((x(z(zy)))z)z)y) \\
&\approx x(y(z(z(x(z(zy)))))) \\
&\overset{(3.8)}{\approx} x(y(x(z(zy)))) \\
&\overset{(3.7)}{\approx} z(z(y(yx))) \\
&\overset{(3.3)}{\approx} z(z(y(y(z(zx))))) \\
&\overset{(3.3)}{\approx} z(z(y(y(z(z(y(yx))))))) \\
&\overset{C15}{\approx} z((((zy)y)(z(z(y(yx))))) \\
&\approx z((y(zy))(z(z(y(yx))))) \\
&\overset{(3.5)}{\approx} z((y(zy))((zy)(x(zy))))
\end{aligned}$$

$$
\begin{aligned}
&\approx z((y(zy))((zy)((zy)x)))\\
&\overset{C15}{\approx} z((((y(zy))(zy))(zy))x)\\
&\overset{C15}{\approx} z((y((zy)((zy)(zy))))x)\\
&\approx z((y(zy))x)\\
&\approx z(((zy)y)x)\\
&\overset{C15}{\approx} z(z(y(yx)))\\
&\approx z((y(xy))z)\\
&\approx z((x \vee y)z)\\
&\approx (x \vee y) \vee z
\end{aligned}
$$

(P3) $x \vee (y \vee z) \approx x \vee (z \vee y)$:

$$
\begin{aligned}
x \vee (y \vee z) &\approx (y \vee z)(x(y \vee z))\\
&\approx (z(yz))(x(z(yz)))\\
&\approx ((z(yz))(x(z(yz))))((z(yz))(x(z(yz))))\\
&\approx ((x(z(yz)))(z(yz)))((z(yz))(x(z(yz))))\\
&\overset{C15}{\approx} x((z(yz))((z(yz))((z(yz))(x(z(yz))))))\\
&\overset{C15}{\approx} x((((z(yz))(z(yz)))(z(yz)))(x(z(yz))))\\
&\approx x((z(yz))(x(z(yz))))\\
&\approx x(((yz)z)(x(z(zy))))\\
&\overset{C15}{\approx} x(y(z(z(x(z(zy))))))\\
&\overset{(3.8)}{\approx} x(y(x(z(zy))))\\
&\overset{(3.7)}{\approx} z(z(y(yx)))\\
&\overset{(3.13)}{\approx} (y(zy))(x(y(zy)))\\
&\approx x \vee (y(zy))\\
&\approx x \vee (z \vee y)
\end{aligned}
$$

(P4) $x \vee (yz) \approx x \vee (y \vee z)$:

$$\begin{aligned}
x \vee (yz) &\approx (yz)(x(yz)) \\
&\overset{(3.11)}{\approx} z(z(y(yx))) \\
&\overset{(3.12)}{\approx} y(y(z(zx))) \\
&\overset{(3.13)}{\approx} (z(yz))(x(z(yz))) \\
&\approx x \vee (z(yz)) \\
&\approx x \vee (y \vee z)
\end{aligned}$$

□

BIBLIOGRAPHY

[1] Reza Akhtar, Ashley Arp, Michael Kaminski, Jasmine Van Exel, Davian Vernon, and Cory Washington. The varieties of Bol-Moufang quasigroups defined by a single operation. *Quasigroups Related Systems*, 20(1):1–10, 2012.

[2] Libor Barto and Marcin Kozik. Constraint satisfaction problems of bounded width. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009)*, pages 595–603. IEEE Computer Soc., Los Alamitos, CA, 2009.

[3] Clifford Bergman. *Universal Algebra*, volume 301 of *Pure and Applied Mathematics (Boca Raton)*. CRC Press, Boca Raton, FL, 2012.

[4] Garrett Birkhoff. On the structure of abstract algebras. *Proceedings of the Cambridge Philosophical Society*, 31:433–454, 1935.

[5] Garrett Birkhoff. Subdirect unions in universal algebra. *Bull. Amer. Math. Soc.*, 50:764–768, 1944.

[6] Andrei Bulatov. Combinatorial problems raised from 2-semilattices. *J. Algebra*, 298(2):321–339, 2006.

[7] Andrei Bulatov and Víctor Dalmau. A simple algorithm for Mal′tsev constraints. *SIAM J. Comput.*, 36(1):16–27 (electronic), 2006.

[8] Andrei Bulatov and Peter Jeavons. Algebraic structures in combinatorial problems. Technical report, Technische Universitat Dresden, 2001. MATH–AL–4–2001, http://www.cs.sfu.ca/~abulatov/papers/varieties.ps.

[9] Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.*, 34(3):720–742 (electronic), 2005.

[10] Andrei Bulatov and Matthew Valeriote. Recent results on the algebraic approach to the CSP. In Nadia Creignou, Phokion G. Kolaitis, and Heribert Vollmer, editors, *Complexity of Constraints*, pages 68–92. Springer-Verlag, Berlin, Heidelberg, 2008.

[11] Stanley Burris and H. P. Sankappanavar. *A course in universal algebra*, volume 78 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1981.

[12] B. Coté, B. Harvill, M. Huhn, and A. Kirchman. Classification of loops of generalized Bol-Moufang type. *Quasigroups Related Systems*, 19(2):193–206, 2011.

[13] Víctor Dalmau. Generalized majority-minority operations are tractable. *Log. Methods Comput. Sci.*, 2(4):4:1, 14, 2006.

[14] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: a study through Datalog and group theory. *SIAM J. Comput.*, 28(1):57–104 (electronic), 1999.

[15] Ralph Freese, Emil Kiss, and Matthew Valeriote. Universal Algebra Calculator, 2011. Available at: `www.uacalc.org`.

[16] Joan Hart and Kenneth Kunen. Single axioms for odd exponent groups. *J. Automat. Reason.*, 14(3):383–412, 1995.

[17] Barbara Hernandez, Lauren Morey, and Alicia Velek. Generalized Bol-Moufang identities of loops and quasigroups. Technical report, Miami University, 2012. http://www.units.muohio.edu/sumsri/sumj/2012/algebra_hmv.pdf.

[18] Paweł Idziak, Petar Marković, Ralph McKenzie, Matthew Valeriote, and Ross Willard. Tractability and learnability arising from algebras with few subpowers. *SIAM J. Comput.*, 39(7):3023–3037, 2010.

[19] Peter Jeavons. On the algebraic structure of combinatorial problems. *Theoret. Comput. Sci.*, 200(1-2):185–204, 1998.

[20] Peter Jeavons, David Cohen, and Marc Gyssens. Closure properties of constraints. *J. ACM*, 44(4):527–548, 1997.

[21] Jaroslav Ježek and Tomáš Kepka. The lattice of varieties of commutative abelian distributive groupoids. *Algebra Universalis*, 5(2):225–237, 1975.

[22] Jaroslav Ježek, Tomáš Kepka, and Petr Němec. Distributive groupoids. *Rozpravy Československé Akad. Věd Řada Mat. Přírod. Věd*, 91(3):94, 1981.

[23] K.A. Kearnes and E.W. Kiss. *The shape of congruence lattices*. Memoirs of the American Mathematical Society Series. American Mathematical Society, 2013.

[24] Tomáš Kepka. Commutative distributive groupoids. *Acta Univ. Carolin.—Math. Phys.*, 19(2):45–58, 1978.

[25] Tomáš Kepka and Petr Němec. Commutative Moufang loops and distributive groupoids of small orders. *Czechoslovak Math. J.*, 31(106)(4):633–669, 1981.

[26] Marcin Kozik, Andrei Krokhin, Matthew A. Valeriote, and Ross Willard. Characterizations of several Maltsev conditions. *Preprint*, 2013.

[27] H. Lakser, R. Padmanabhan, and C. R. Platt. Subdirect decomposition of Płonka sums. *Duke Math. J.*, 39:485–488, 1972.

[28] Benoit Larose and László Zádori. The complexity of the extendibility problem for finite posets. *SIAM J. Discrete Math.*, 17(1):114–121 (electronic), 2003.

[29] Benoit Larose and László Zádori. Bounded width problems and algebras. *Algebra Universalis*, 56(3-4):439–466, 2007.

[30] Alan Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1):99–118, 1977. Reprinted in *Readings in Artificial Intelligence*, B. L. Webber and N. J. Nilsson (eds.), Tioga Publ. Col., Palo Alto, CA, pp. 69-78, 1981.

[31] A. I. Maltsev. On the general theory of algebraic systems. *Mat. Sb. N.S.*, 35(77):3–20, 1954.

[32] A. I. Maltsev. Multiplication of classes of algebraic systems. *Sibirsk. Mat. Ž.*, 8:346–365, 1967.

[33] Miklós Maróti. Tree on top of Maltsev. Technical report, University of Szeged, 2010. http://www.math.u-szeged.hu/~mmaroti/pdf/200x%20Tree%20on%20top%20of%20Maltsev.pdf.

[34] Miklós Maróti. Maltsev on top. Technical report, University of Szeged, 2011. http://www.math.u-szeged.hu/~mmaroti/pdf/200x%20Maltsev%20on%20top.pdf.

[35] Miklós Maróti and Ralph McKenzie. Existence theorems for weakly symmetric operations. *Algebra Universalis*, 59(3-4):463–489, 2008.

[36] W. McCune. Prover9 and Mace4. http://www.cs.unm.edu/~mccune/prover9/, 2005–2010.

[37] I. I. Mel′nik. Normal closures of perfect varieties of universal algebras. In *Ordered sets and lattices, No. 1 (Russian)*, pages 56–65. Izdat. Saratov. Univ., Saratov, 1971.

[38] J. D. Phillips and Petr Vojtěchovský. The varieties of loops of Bol-Moufang type. *Algebra Universalis*, 54(3):259–271, 2005.

[39] J. D. Phillips and Petr Vojtěchovský. The varieties of quasigroups of Bol-Moufang type: an equational reasoning approach. *J. Algebra*, 293(1):17–33, 2005.

[40] J. Płonka. On a method of construction of abstract algebras. *Fund. Math.*, 61:183–189, 1967.

[41] J. Płonka. On equational classes of abstract algebras defined by regular equations. *Fund. Math.*, 64:241–247, 1969.

[42] Robert W. Quackenbush. Varieties of Steiner loops and Steiner quasigroups. *Canad. J. Math.*, 28(6):1187–1198, 1976.

[43] Anna Romanowska. On regular and regularised varieties. *Algebra Universalis*, 23(3):215–241, 1986.

[44] Anna. Romanowska and Jonathan D. H. Smith. *Modes*. World Scientific Publishing Co. Inc., River Edge, NJ, 2002.

[45] Thomas J. Schaefer. The complexity of satisfiability problems. In *Conference Record of the Tenth Annual ACM Symposium on Theory of Computing (San Diego, Calif., 1978)*, pages 216–226. ACM, New York, 1978.

[46] David Stanovský. Distributive groupoids are symmetric-by-medial: an elementary proof. *Comment. Math. Univ. Carolin.*, 49(4):541–546, 2008.

[47] Walter Taylor. Varieties obeying homotopy laws. *Canad. J. Math.*, 29(3):498–527, 1977.