MAL'TSEV CONDITION SATISFACTION PROBLEMS

ON THE COMPLEXITY OF SEVERAL MAL'TSEV CONDITION SATISFACTION PROBLEMS

By
J P ROONEY, M.MATH

A Thesis Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements for the Degree
Doctor of Philosophy

McMaster University DOCTOR OF PHILOSOPHY (2020), Hamilton, Ontario (Mathematics)

TITLE: On the Complexity of Several Mal'tsev Condition Satisfaction Problems
AUTHOR: J P Rooney, MMath (University of Bath)
SUPERVISOR: Professor Matthew Anthony Valeriote
NUMBER OF PAGES: vii, 98

Abstract

In this thesis we derive novel results on the complexity of idempotent Mal'tsev condition satisfaction problems. For a Mal'tsev condition Σ, the idempotent Σ-satisfaction problem is the decision problem defined via:

- INPUT: A finite idempotent algebra $\mathbf{A}$.
- QUESTION: Does $\mathbf{A}$ satisfy Σ?

In particular we are able to prove that this decision problem is in the complexity class NP whenever Σ satisfies one of the following conditions:

1. Σ is a strong Mal'tsev condition which implies the existence of a near unanimity term.
2. Σ is a strong Mal'tsev condition of height < 1 (see Definition 5.1.1).

As a porism of these two results, we are able to derive the stronger result that the complexity of the idempotent Σ-satisfaction problem is in NP whenever Σ is a strong Mal'tsev condition which implies the existence of an edge term.

On top of this we also outline a polynomial-time algorithm for the idempotent Σ-satisfaction problem when Σ is a linear strong Mal'tsev condition which implies the existence of a near unanimity term.

We also examine the related search problem in which the goal is to produce operation tables of term operations of the algebra $\mathbf{A}$ which witness that $\mathbf{A}$ satisfies the Mal'tsev condition Σ whenever such terms exist (and otherwise correctly decide that such terms do not exist). We outline polynomial-time algorithms for this search problem for various strong Mal'tsev conditions.

We close the thesis with a short list of open problems as suggested directions for further research.

Acknowledgements

The writing of this thesis would not have been possible without the immense and unfathomable support of my wonderful partner Michelle Alonso de Mesa. I am eternally grateful for the strength you lend to all my endeavours.

I would also like to acknowledge my supervisor Matt Valeriote whose expert guidance and endless patience have paved the way for my novel contributions to this fascinating field of study. I could not have asked for a superior mentor.

To the logic group at McMaster I also extend my sincere appreciation. Particular thanks to Dr Bradd Hart, Dr Deirdre Haskell, and Dr Patrick Speissegger for your valued contribution to my mathematical education. Special thanks also to my PhD brother Dr Alberto Chicco, whose time at McMaster was altogether too short.

To the many other friends and family members who have enriched my life I also extend my deep gratitude. Each of you has contributed to this thesis in countless and unmeasurable ways. Thank you.

There once was a lad from McMaster
Whose Thesis turned out a disaster
The results stated here
Are rather unclear
But perhaps you can see what I'm after

Contents

List of Figures

Introduction

This thesis concerns itself with Mal'tsev condition satisfaction problems (MCSPs). We view this field of study as a natural extension of research on the algebraic approach to constraint satisfaction problems (CSPs) which was initiated by Jeavons [27] in 1998 and recently led to the successful resolution of Feder and Vardi's [17] much-investigated CSP dichotomy conjecture with the publications [14] and [44]. The CSP dichotomy theorem now clearly delineates those templates (or algebras) which give rise to tractable subclasses of the CSP and those which give rise to NP-complete subclasses. The dividing line can be characterized as the satisfaction of a particular nontrivial Mal'tsev condition.

The questions naturally arising are what Chen and Larose call "the metaquestions in constraint tractability" [16] and can be asked (as in Chen and Larose's work [16]) from the perspective of templates (relational structures) or (as in [20, 24, 30, 29, 19]) from the algebraic perspective (where the focus is on algebraic structures, or *algebras* for short).

We focus here on the algebraic version of these "metaquestions" which we call Mal'tsev condition satisfaction problems. For a fixed Mal'tsev condition Σ the Σ-satisfaction problem is the following decision problem:

- INPUT: A finite algebra $\mathbf{A}$.
- QUESTION: Does $\mathbf{A}$ satisfy Σ?

As an example, consider the Mal'tsev condition of having a binary idempotent cyclic term. That is, a term which satisfies $b(x, y) \approx b(y, x)$ and $b(x, x) \approx x$. Given a finite algebra $\mathbf{A}$ we can determine whether or not $\mathbf{A}$ satisfies this Mal'tsev condition by simply building all of the binary term operations of $\mathbf{A}$ and checking to see whether any satisfy these two equations. It is well-known to universal algebraists that such an algorithm can be executed in exponential-time with respect to the size of the algebra $\mathbf{A}$. It follows from [24, Theorem 3.5] that this problem is indeed EXPTIME-complete. On the other hand, it follows from [4, Lemma 2.4] that the following variant of the problem is in the class P:

- INPUT: A finite idempotent algebra $\mathbf{A}$.
- QUESTION: Does $\mathbf{A}$ have a binary idempotent cyclic term?

This is what we call the idempotent Mal'tsev condition satisfaction problem (for the particular Mal'tsev condition of having a binary idempotent cyclic term) and the results of [20] provide several other examples of Mal'tsev conditions which have EXPTIME-complete satisfaction problems and tractable idempotent satisfaction problems. It is worth noting at this point that applications of these results are not limited to the field of constraint satisfaction. In fact many tractability results arising in this area find practical applications as research tools for universal algebraists and some of the algorithms developed have been successfully implemented as part of the calculation software UACalc [18].

The main goal of this thesis is to explore the complexity of MCSPs for several Mal'tsev conditions of interest to universal algebraists. Inspired by the results of [19] we were interested in exploring the question: do nonlinear Mal'tsev conditions give rise to harder decision problems than linear Mal'tsev conditions? Given a Mal'tsev condition Σ, one of the primary techniques used to establish a tractability result for the idempotent Σ-satisfaction problem is to compose together term operations which satisfy the equations of Σ on different subsets of the algebra to obtain an operation which satisfies the equations globally. This technique works well in a number of situations (see for example [24, Theorem 2.6], [42, Theorem 2.2], [30, Theorem 7]) all of which involve linear Mal'tsev conditions but it seems unlikely that this type of proof can work when the Mal'tsev condition Σ is nonlinear. The results of our explorations are laid out in this thesis as described in the remainder of this introduction.

In Chapter 1 we begin by introducing all the relevant definitions from universal algebra and in Section 1.2 we provide a description of those Mal'tsev conditions which feature in this text, including some motivational remarks. In Chapter 2 we introduce the relevant definitions from the field of complexity theory, including in Section 2.2 an introduction to the problems CSP and SMP and then in Section 2.3 we formally define Mal'tsev condition satisfaction problems and explore existing results in the field.

In Chapter 3 we begin to explore novel results obtained in this thesis. In Section 3.1 we use the work of Baker and Pixley [2] to derive a tractability result in the idempotent case for linear Mal'tsev conditions which imply the existence of near unanimity terms. In Section 3.2 we are able to use the same kind of reasoning to determine the existence of nonlinear Mal'tsev conditions whose idempotent satisfaction problem is in the class NP. This represents the first example of a nonlinear Mal'tsev condition whose idempotent satisfaction problem is not EXPTIME-complete (assuming NP $\neq$ EXPTIME) which is perhaps an indication that the hypothesis that nonlinear conditions are necessarily hard to detect may be incorrect.

We take an excursion in Chapter 4 to turn our attention to the related search problem associated to MCSPs. In their exploration of the idempotent minority satisfaction problem the authors of [29] derive the related result that obtaining term operations witnessing the satisfaction of the existence of a Mal'tsev term[1] in a finite idempotent algebra is a tractable search problem [29, Corollary 7]. We reproduce this

[1] "Existence of a Mal'tsev term" is the name of a Mal'tsev condition. See Example 1.2.6.

result in Section 4.3 and in Section 4.4 we use the result of [24, Theorem 2.6] to extend this to a much broader class of conditions. Sections 4.2 and 4.5 contain similar results for the conditions existence of a k-ary cyclic term and congruence n-permutability respectively.

In Chapter 5 we return to the consideration of the decision problem MCSP. We show that conditions of height < 1 in the sense described in [6, Definition 5.1] (formally defined here in Definition 5.1.1) give rise to idempotent satisfaction problems which are also in the class NP. In particular, this extends the result for minority terms [29, Theorem 17] to include all cube term conditions.

Finally, in Chapter 6 we summarize all the results obtained in this thesis and develop a common generalization of Theorems 3.2.1 and 5.2.4. Namely, we show (Theorem 6.1.4) that any strong Mal'tsev condition which implies the existence of an edge term gives rise to an idempotent satisfaction problem in the complexity class NP. We then suggest some open problems as direction for future research (Section 6.2).

Chapter 1

On the Basics of Universal Algebra

In this chapter we introduce those basic notions of universal algebra which are necessary for an understanding of the results found in subsequent chapters of this thesis (and their proofs). A more thorough introduction to the topics discussed here can be found in any universal algebra textbook (see for example [15, 22, 7]). Readers already familiar with the field may wish to skip this chapter and refer back to the definitions herein as necessary.

1.1 Algebras, Terms, Varieties, and Free Algebras

There are two main players in the great drama of this thesis. The first major player is the algebra.

Definition 1.1.1. An *algebra* $\mathbf{A} := (A, \mathcal{F})$ is given by a nonempty set A together with an indexed set $\mathcal{F} := \{f_i^{\mathbf{A}} \mid i \in I\}$ of finitary operations on A. The set A is called the *universe* of the algebra $\mathbf{A}$ and every $f \in \mathcal{F}$ is called a *basic operation of* $\mathbf{A}$. The function $\rho_{\mathbf{A}} : I \to \mathbb{N}$ which assigns to each $i \in I$ the arity of the basic operation $f_i^{\mathbf{A}}$ of $\mathbf{A}$ is called the *similarity type of* $\mathbf{A}$. The superscript in $f_i^{\mathbf{A}}$ is useful particularly when we are thinking of the index set I as a collection of function symbols (in which case $f \in I$ is a function symbol of arity $\rho(f)$ whereas $f^{\mathbf{A}}$ is the corresponding operation on A of the same arity). In practice we omit the superscript when we think this will not cause any confusion. An algebra is said to be *finite* if the sets A and $\mathcal{F}$ are both finite. In this case we define the *size* of the algebra $\mathbf{A}$ to be the natural number $\sum_{i \in I} |A|^{\text{arity}(f_i^{\mathbf{A}})}$ which we denote by $||\mathbf{A}||$.[1] An operation g on A is said to be *idempotent* if it satisfies $g(a, a, \ldots, a) = a$ for each $a \in A$. The algebra $\mathbf{A}$ is said to be *idempotent* if each basic operation of $\mathbf{A}$ is idempotent.

Example 1.1.2.

- A group $\mathbf{G} := (G, \{f_1, f_2, f_3\})$ is an algebra of similarity type $\rho_{\mathbf{G}} : \{1, 2, 3\} \to \mathbb{N}$ defined by $\rho_{\mathbf{G}}(1) = 2,\ \rho_{\mathbf{G}}(2) = 1,\ \rho_{\mathbf{G}}(3) = 0$. In this case we

[1] Throughout this thesis we assume that every algebra has at least one basic operation of arity at least one, so that $|A| \leq ||\mathbf{A}||$.

are defining f_1 to be the multiplication of the group, f_2 is the inverse (considered as a unary operation $x \mapsto x^{-1}$), and f_3 is the nullary operation whose constant value is the identity of $\mathbf{G}$. We note that the similarity type of a group is usually abbreviated as $(2, 1, 0)$, omitting an explicit description of the indexing set I (and similarly for other algebras with finitely many basic operations).

- Similarly, a ring (with unit) $\mathbf{R} := (R, \cdot, +, -, 1, 0)$ is an algebra of similarity type $(2, 2, 1, 0, 0)$.
- A lattice $\mathbf{L} := (L, \wedge, \vee)$ is an algebra of similarity type $(2, 2)$.
- A Boolean algebra is an algebra $\mathbf{B} := (B, \wedge, \vee, \neg, \top, \bot)$ of similarity type $(2, 2, 1, 0, 0)$.

On many occasions we will be required to consider various *powers* of a given algebra. The definition is as follows:

Definition 1.1.3. Given a family $(\mathbf{A}_j)_{j\in J}$ of algebras all of the same similarity type $\rho : I \to \mathbb{N}$, we define the *direct product of* $(\mathbf{A}_j)_{j\in J}$, denoted by $\prod_{j\in J} \mathbf{A}_j$, to be the algebra $\mathbf{P}$ whose underlying set is the Cartesian product $P := \prod_{j\in J} A_j$ of the sets A_j and such that the operation f_i acts "coordinate-wise" on the elements of the product. Formally, an element $p \in \prod_{j\in J} A_j$ is a function $p : J \to \bigcup_{j\in J} A_j$ such that for each $j \in J$ we have $p(j) \in A_j$. Given an index $i \in I$ the operation $f_i^{\mathbf{P}}$ is given by defining for each $p_1, \ldots, p_{\rho(i)} \in P$:

$$f_i^{\mathbf{P}}(p_1, \ldots, p_{\rho(i)})(j) := f_i^{\mathbf{A}_j}(p_1(j), \ldots, p_{\rho(i)}(j)) \tag{1.1}$$

For any set X we can similarly define the algebra $\mathbf{A}^X$ to be the product algebra $\prod_{x\in X} \mathbf{A}_x$ where $\mathbf{A}_x := \mathbf{A}$ for all $x \in X$. The elements of A^X are therefore all of the functions from X to A and the basic operations of $\mathbf{A}$ act on A^X via the natural action on the image points as defined in 1.1. Let $[n]$ denote the subset $\{1, \ldots, n\} \subset \mathbb{N}$. We typically write $\mathbf{A}^n$ and A^n for $\mathbf{A}^{[n]}$ and $A^{[n]}$ respectively and we typically write the elements of A^n as "tuples" $(a_1, \ldots, a_n)$. Formally, $(a_1, \ldots, a_n)$ is used to denote the function $f : [n] \to A : k \mapsto a_k$. The algebra $\mathbf{A}^n$ is called the *n-th (direct) power of* $\mathbf{A}$.

Given a tuple $(a_1, \ldots, a_n) \in A^n$ and a subset $K \subseteq [n]$ we define $(a_1, \ldots, a_n)|_K$ to be the restriction of the tuple $(a_1, \ldots, a_n)$ to the subset K. If $K = \{k\}$ is a one-element subset of $[n]$ then we write $(a_1, \ldots, a_n)|_k$ in place of $(a_1, \ldots, a_n)|_K$. Whereas formally $(a_1, \ldots, a_n)|_k$ is a function $f \in A^{\{k\}}$ with $f(k) = a_k$ we will always identify $(a_1, \ldots, a_n)|_k$ with the element $a_k \in A$.

Example 1.1.4. The familiar examples of direct products of (e.g.) groups, rings, lattices are all examples of this more general direct product. We can also create

product algebras from algebras in distinct classical classes when the similarity types are the same. For example let $\mathbb{Z}_2 := (\{0,1\}, \cdot, +, -, 1, 0)$ denote the ring of integers modulo 2 and $\mathbf{2} := (\{0,1\}, \wedge, \vee, \neg, \top, \bot)$ denote the two element chain (considered as a Boolean algebra).

The algebra $\mathbf{P} := \mathbb{Z}_2 \times \mathbf{2}$ has similarity type $(2,2,1,0,0)$ (as do $\mathbb{Z}_2$ and $\mathbf{2}$) and a standard application of the basic operations of $\mathbf{P}$ is given by (e.g.):

$$f_1^{\mathbf{P}}((0,0),(1,0)) = (f_1^{\mathbb{Z}_2}(0,1), f_1^{\mathbf{2}}(0,0)) = (0 \cdot_{\mathbb{Z}_2} 1, 0 \wedge_{\mathbf{2}} 0) = (0,0)$$

Aside from powers of an algebra we will also need to consider *subalgebras* and *subpowers*. The relevant definitions are:

Definition 1.1.5. Let $\mathbf{A} := (A, \mathcal{F})$ and $\mathbf{B} := (B, \mathcal{G})$ be algebras of the same similarity type $\rho : I \to \mathbb{N}$. We say that $\mathbf{B}$ *is a subalgebra of* $\mathbf{A}$ if $B \subseteq A$ and for each $g_i \in \mathcal{G}$ we have $g_i = f_i|_B$ (where $f_i|_B$ denotes the restricted function $f_i|_B : B^{\rho(i)} \to A : \bar{b} \mapsto f_i(\bar{b})$). It follows that for each basic operation f_i of A, the set B is *closed under* f_i. I.e. if $\bar{b} \in B^{\rho(i)}$ is a tuple of elements from B, then $f_i(\bar{b})$ is an element of B. Any subset $X \subseteq A$ with this property (being closed under all the basic operations of $\mathbf{A}$) is called a *subuniverse* of $\mathbf{A}$. Every nonempty subuniverse therefore corresponds to a subalgebra of $\mathbf{A}$ by equipping it with the restrictions of the basic operations of $\mathbf{A}$ (which are necessarily well-defined operations on the subuniverse). We write $\mathbf{B} \leq \mathbf{A}$ to mean $\mathbf{B}$ is a subalgebra of $\mathbf{A}$. We call an algebra $\mathbf{C}$ a *subpower of* $\mathbf{A}$ if there is some set X such that $\mathbf{C} \leq \mathbf{A}^X$.

Example 1.1.6. The ring $\mathbb{R}^{\mathbb{R}}$ is the ring of real-valued functions on domain $\mathbb{R}$ under point-wise multiplication and addition. The ring $C(\mathbb{R})$ of continuous real-valued functions on domain $\mathbb{R}$ is therefore a subpower of the ring $\mathbb{R}$ (since it is a subalgebra of $\mathbb{R}^{\mathbb{R}}$).

It is an elementary exercise in universal algebra to show that for any algebra $\mathbf{A}$ and any set $(B_\lambda)_{\lambda \in \Lambda}$ of subuniverses of $\mathbf{A}$, the intersection $B := \bigcap_{\lambda \in \Lambda} B_\lambda$ is also a subuniverse of $\mathbf{A}$. Hence we are justified in making the following definition:

Definition 1.1.7. Let $\mathbf{A}$ be an algebra. Given elements $a_1, \ldots, a_n \in A$ we define the *subalgebra of* $\mathbf{A}$ *generated by* $a_1, \ldots, a_n$ to be the smallest subalgebra of $\mathbf{A}$ whose universe contains all of the elements $a_1, \ldots, a_n$. We denote this algebra by $\langle a_1, \ldots, a_n \rangle_{\mathbf{A}}$ omitting the subscript $_{\mathbf{A}}$ whenever we think that this will not lead to any confusion.

Whereas we have taken the decision to frame the results of this thesis as results concerning finite algebras, many of these results (e.g. Theorems 3.1.8, 3.2.1, 5.2.4) could equally well be thought of as results concerning the "varieties generated by" finite algebras satisfying the relevant conditions of each result. In many of the proofs we will also have cause to reference "free algebras" in certain "varieties" and other related notions. The relevant definitions close off this section but the reader is referred to again to [15, 7, 22] for a more detailed introduction to these objects of study.

Definition 1.1.8. Let $\mathbf{A} = (A, \mathcal{F})$ and $\mathbf{B} = (B, \mathcal{G})$ be algebras of the same similarity type $\rho : I \to \mathbb{N}$. A function $h : A \to B$ is called a *homomorphism* if for each $i \in I$ and each $a_1, \ldots, a_{\rho(i)} \in A$ we have $h\left(f_i(a_1, \ldots, a_{\rho(i)})\right) = g_i\left(h(a_1), \ldots, h(a_{\rho(i)})\right)$. We say that $\mathbf{B}$ is a *homomorphic image* of $\mathbf{A}$ if there is a surjective homomorphism $h : A \twoheadrightarrow B$. We say that $\mathbf{A}$ and $\mathbf{B}$ are *isomorphic* algebras if there is an invertible homomorphism from $\mathbf{A}$ to $\mathbf{B}$ (whose inverse is necessarily an invertible homomorphism from $\mathbf{B}$ to $\mathbf{A}$).

Example 1.1.9. The function $h : \mathbb{Z}_9 \to \mathbb{Z}_{12}$ defined via $h([k]_9) := [4k]_{12}$ is a homomorphism of additive groups (considered as algebras of type $(2, 1, 0)$). The image of the homomorphism is the subalgebra $(\{[0]_{12}, [4]_{12}, [8]_{12}\}, +, -, 0)$ which is therefore a homomorphic image of $\mathbb{Z}_9$.

Given a class K of algebras all of the same simlarity type ρ, we may define three new classes:

- $H(K)$: the class of all algebras $\mathbf{C}$ which are a homomorphic image of some algebra $\mathbf{A}$ in K
- $S(K)$: the class of all algebras $\mathbf{C}$ which are isomorphic to a subalgebra $\mathbf{B}$ of some algebra $\mathbf{A}$ in K
- $P(K)$: the class of all algebras $\mathbf{C}$ which are isomorphic to a direct product $\prod_{j \in J} \mathbf{A}_j$ of algebras $\mathbf{A}_j$ in K

Definition 1.1.10. A *variety* $\mathcal{V}$ is a class of algebras all of the same similarity type which is closed under the three operators H, S and P defined above.

Example 1.1.11. The class of all groups is clearly a variety, since the definition of homomorphism (and hence of homomorphic image) matches the usual definition of group homomorphism, the definition of subalgebra matches that of subgroup, and direct products correspond with the usual direct products of groups. Standard results from group theory tell us that all of these constructions constitute groups, and hence the class is closed under the operators defined above.

It should be clear that given a family $(V_\lambda)_{\lambda \in \Lambda}$ of varieties of the same similarity type, the intersection $V := \bigcap_{\lambda \in \Lambda} V_\lambda$ is also a variety (of the same similarity type). It should also be clear that the class of all algebras of some fixed similarity type is a variety. This justifies the following definition:

Definition 1.1.12. Given a class K of algebras of the same similarity type, we define the *variety generated by* K, $V(K)$, to be the smallest variety containing all the members of the class K. A variety $\mathcal{V}$ is *finitely generated* if there is a finite set of finite algebras K such that $\mathcal{V} = V(K)$.

A fundamental theorem of Birkhoff [9] tells us that varieties are characterized by the equations which they satisfy. Before we can concretize this we need a few more definitions- some of which are also necessary in outlining the Mal'tsev Condition Satisfaction Problem (defined in Chapter 2) which is the central topic of this thesis. We begin by defining "terms" of an algebra, and of a variety.

Definition 1.1.13. Let $\rho : I \to \mathbb{N}$ be a similarity type of algebras and X a set which is disjoint from I whose elements we will call *variables*. We recursively define the *terms of type ρ over X* as follows:

- If $x \in X$ or $x \in \rho^{-1}(\{0\})$, then x is a term of type ρ over X, and
- If $t_1, \ldots, t_n$ are terms of type ρ over X and $f \in \rho^{-1}(\{n\})$, then $f(t_1, \ldots, t_n)$ is a term of type ρ over X.

Given an algebra $\mathbf{A} := (A, \mathcal{F})$ of similarity type ρ we define the *n-ary terms of $\mathbf{A}$* (for n in $\mathbb{N}$) to be the terms of type ρ over a set $\{x_1, \ldots, x_n\}$ of size n disjoint from A and I. By a *term of $\mathbf{A}$* we mean an n-ary term of $\mathbf{A}$ for some $n \in \mathbb{N}$. If $\mathcal{V}$ is a variety of similarity type ρ, then an *n-ary term of $\mathcal{V}$* is simply a term of type ρ over a set of variables of size n and a *term of $\mathcal{V}$* is an n-ary term of $\mathcal{V}$ for some $n \in \mathbb{N}$.

Example 1.1.14. If we fix a similarity type $\rho : \{\cdot, ^{-1}, 1\} \to \mathbb{N}$ for the variety of all groups (hence $\rho(\cdot) = 2,\ \rho(^{-1}) = 1$, and $\rho(1) = 0$), then we see that the expressions: $(x \cdot (y^{-1})) \cdot z$ and $x \cdot ((y^{-1}) \cdot z)$ are two (different) terms of groups. Here we have used familiar infix notation whereas formally the expressions ought to be $\cdot(\cdot(x, ^{-1}(y)), z)$ and $\cdot(x, \cdot(^{-1}(y), z))$. We will prefer the former whenever we believe it is unlikely to cause confusion.

The eagle-eyed reader will have noticed that the terms of groups introduced in the previous example ought to be "equivalent" in some sense due to the associative law satisfied by every group. This notion of equivalence is described by saying that the two terms "induce the same term operation on $\mathbf{G}$" whenever $\mathbf{G}$ is a group. We first define "induced term operations" and then we will define "term algebras" over a fixed similarity type and then "free algebras" which will be quotients of term algebras under the equivalence relation given by this definition of equivalence.

Definition 1.1.15. Let $\mathbf{A} = (A, \mathcal{F})$ be an algebra of similarity type ρ and let t be an n-ary term of $\mathbf{A}$ (over the set $X := \{x_1, \ldots, x_n\}$). We recursively define the *term operation on $\mathbf{A}$ induced by t*, denoted $t^{\mathbf{A}}$, as follows:

- If $t \in I$, then the term operation on $\mathbf{A}$ induced by t is $t^{\mathbf{A}} \in \mathcal{F}$ (the basic operation of $\mathbf{A}$ which is indexed by $t \in I$)
- If $t = x_j \in X$, then the term operation on $\mathbf{A}$ induced by t is $\pi_j^{\mathbf{A}}(x_1, \ldots, x_n)$, the j-th n-ary projection on $\mathbf{A}$ defined via $\pi_j^{\mathbf{A}}(a_1, \ldots, a_n) = a_j$ for all $a_1, \ldots, a_n \in A$

- If $t = f(t_1, \dots, t_k)$ for some $f \in \rho^{-1}(\{k\})$ and some n-ary terms $t_1, \dots, t_k$ of $\mathbf{A}$, then the term operation on $\mathbf{A}$ induced by t is

$$f^{\mathbf{A}}(t_1^{\mathbf{A}}(x_1, \dots, x_n), \dots, t_k^{\mathbf{A}}(x_1, \dots, x_n))$$

  where $f^{\mathbf{A}}$ is the basic operation of $\mathbf{A}$ indexed by $f \in I$ and $t_i^{\mathbf{A}}(x_1, \dots, x_n)$ is the term operation on $\mathbf{A}$ induced by the n-ary term t_i of $\mathbf{A}$ (for each $1 \leq i \leq k$).

Example 1.1.16. It is easy to calculate that the term operation induced on the ring $\mathbb{R}$ by the term $((x_1 + (0 \cdot x_1)) + (-(x_1 \cdot x_2))) + 1$ is the operation defined by $(x, y) \mapsto 1 + x - xy$

Definition 1.1.17. Let $\rho : I \to \mathbb{N}$ be a similarity type of algebras and X a set of variables. An *equation of type ρ over X* is a pair of terms of type ρ over some finite subset $X' \subseteq X$, written $p \approx q$. An algebra $\mathbf{A}$ of similarity type ρ is said to *satisfy* the equation $p \approx q$ if for any $\overline{a} \in A^{X'}$ we have $p^{\mathbf{A}}(\overline{a}) = q^{\mathbf{A}}(\overline{a})$.[2] In this case, $\mathbf{A}$ is called a *model* of the equation $p \approx q$. A class K of algebras is said to *satisfy* the equation $p \approx q$ if every $\mathbf{A}$ in K is a model of $p \approx q$. Given a set Σ of equations of type ρ we say that K satisfies Σ if every algebra in K is a model of every equation in Σ. We use the notation $\mathrm{Mod}(\Sigma)$ to denote the class of all models of Σ, and $\mathrm{Id}(K)$ to denote the class of all equations of type ρ over some fixed countable set $X = \{x_1, x_2, \dots\}$ of variables which are satisfied by every member of K.

The following theorem due to Birkhoff [9] is considered fundamental to the study of universal algebra. It provides a characterization of varieties which is familiar to every universal algebraist and will be implicitly assumed throughout this text.

Theorem 1.1.18. *Let $\mathcal{V}$ be a variety. Then $\mathcal{V}$ =Mod(Id($\mathcal{V}$)). Furthermore, for any class K of algebras of some fixed similarity type, we have:*

$$V(K) = HSP(K) = Mod(Id(K)).$$

We end this section by defining two special algebras of similarity type ρ.

Definition 1.1.19. Let $\rho : I \to \mathbb{N}$ be a similarity type of algebras and X a set of variables (disjoint from I). We define the *term algebra of type ρ over X* to be the algebra $\mathbf{T}_\rho(X)$ whose underlying set is the set of all terms of type ρ over X, denoted $T_\rho(X)$, and whose basic operations are defined as follows: for each $f \in I$ there is an operation $f^{\mathbf{T}_\rho(X)}$ of arity $\rho(f)$ defined via $f^{\mathbf{T}_\rho(X)}(t_1, \dots, t_{\rho(f)}) := f(t_1, \dots, t_{\rho(f)})$ for any choice of terms $t_1, \dots, t_{\rho(f)} \in T_\rho(X)$. Note that $f_i^{\mathbf{T}_\rho(X)}$ is a well-defined operation on $T_\rho(X)$ by the definition of terms of type ρ over X.

[2] There is some ambiguity here which we hope to clear up with an example. If p is a term over the variables x, y and q a term over x, z (for example) given $(a, b, c) \in A^{\{x,y,z\}}$ we regard $p^{\mathbf{A}}((a, b, c))$ as $p^{\mathbf{A}}(a, b)$ and $q^{\mathbf{A}}((a, b, c))$ as $q^{\mathbf{A}}(a, c)$.

The final definition in this section will be that of "free algebras". There will be very few explicit references to free algebras within this thesis and the definition provided here is chosen to reflect the most direct applications of free algebras found in the following pages. Equivalent formulations will be discussed briefly following the definition:

Definition 1.1.20. Let $\mathcal{V}$ be a class of algebras of similarity type ρ and let $\mathbf{T}(X)$ be the term algebra of type ρ over X. Let $\sim_{\mathcal{V}}$ be the equivalence relation on $T(X)$ defined via $s \sim_{\mathcal{V}} t$ if and only if every algebra in $\mathcal{V}$ is a model of the equation $s \approx t$. The algebra $\mathbf{F}_{\mathcal{V}}(X)$ of type ρ, whose underlying set is the set of equivalence classes of terms in $T(X)$ and whose basic operations are defined similarly to those of the term algebra of type ρ over X (via the action of those basic operations on representatives for given classes) is a well-defined algebra called the *free algebra in $\mathcal{V}$ over X*.

The algebra defined above is called "free (in $\mathcal{V}$)" because it satisfies the suggested universal mapping property: namely if $h : X \to A$ is a function from X to the underlying set of some $\mathbf{A}$ in $\mathcal{V}$, then h extends uniquely to a homomorphism $h : \mathbf{F}_{\mathcal{V}}(X) \to \mathbf{A}$. Using the theorem of Birkhoff provided above it can be seen that if $\mathcal{V}$ is $V(\mathbf{A})$ for some algebra $\mathbf{A}$, then $\mathbf{F}_{\mathcal{V}}(\{x_1, \ldots, x_n\})$ is (clearly isomorphic to) the algebra whose underlying set consists of all induced term operations on $\mathbf{A}$ (induced by the terms of the similarity type of $\mathbf{A}$ over the set $\{x_1, \ldots, x_n\}$). This algebra is called the *clone of n-ary term operations on* $\mathbf{A}$ and we recall here that every term operation of $\mathbf{A}$ can be obtained as a composition of the basic operations of $\mathbf{A}$ (see Chapter 4 of [7] or Chapter 10 of [15] for more details).

1.2 Mal'tsev Conditions

The second major player in this thesis is the Mal'tsev Condition. We begin by exploring the definition and then give some examples of Mal'tsev conditions of special import in this thesis and in general.

Definition 1.2.1. A *strong Mal'tsev condition* is a finite set of equations Σ of some type ρ (which we can also take to be finite since we have only finitely many equations). We say that the algebra $\mathbf{A}$ (of type τ possibly distinct from ρ) *satisfies the Mal'tsev condition* Σ if there are terms $t_1, \ldots, t_n$ of $\mathbf{A}$ such that the algebra $\mathbf{A}' := (A, t_1^{\mathbf{A}}, \ldots, t_n^{\mathbf{A}})$ is of type ρ and $\mathbf{A}'$ satisfies all the equations of Σ in the sense of Definition 1.1.17. I.e. for every function symbol p appearing in the equations of Σ, there is a term operation $p^{\mathbf{A}}$ of the algebra $\mathbf{A}$ such that under this correspondence each equation of Σ relates term operations that are equal (as operations on A). A variety $\mathcal{V}$ *satisfies the Mal'tsev condition* Σ if every algebra in $\mathcal{V}$ satisfies the Mal'tsev condition Σ.[3]

[3] In this case the terms $t_1, \ldots, t_n$ can actually be chosen uniformly. This is well-known to universal algebraists and is a consequence of [15, Theorem 11.4].

We say that the (strong) Mal'tsev condition Σ *implies* the (strong) Mal'tsev condition T if for every algebra $\mathbf{A}$ we have that if $\mathbf{A}$ satisfies Σ, then $\mathbf{A}$ satisfies T. We say that two (strong) Mal'tsev conditions are *equivalent* if each implies the other.

We define *Mal'tsev condition* to mean a sequence $(\Sigma_n)_{n\in\mathbb{N}}$ of strong Mal'tsev conditions Σ_n such that for each $k \in \mathbb{N}$ we have Σ_k implies Σ_{k+1}.[4] An algebra $\mathbf{A}$ (resp. variety $\mathcal{V}$) *satisfies the Mal'tsev condition* $(\Sigma_n)_{n\in\mathbb{N}}$ if $\mathbf{A}$ (resp. every $\mathbf{A}$ in $\mathcal{V}$) satisfies Σ_k for some $k \in \mathbb{N}$.

We say that a (strong) Mal'tsev condition is *nontrivial* if it is not satisfied in the variety of sets, and *consistent* if it does not imply the strong Mal'tsev condition $\{x \approx y\}$. A (strong) Mal'tsev condition is called *linear* if it is equivalent to a Mal'tsev condition which includes only linear equations[5].

We now look at a series of examples. Each example included here has its own importance to the field of universal algebra which we touch upon only briefly when introducing these examples. References are given in each case for further reading. The order in which the examples appear is loosely based on the order in which these conditions are used or discussed throughout the rest of the thesis. The study of Mal'tsev conditions is vast and deep but we limit ourselves here to brief motivations for studying particular conditions and relevant results that will be used later on.

Example 1.2.2. Let k be an integer with $k \geq 3$. The first strong Mal'tsev condition considered in this paper is the condition of having a k-ary near unanimity term. We say that the algebra $\mathbf{A}$ *has a k-ary near unanimity term* if it satisfies the strong Mal'tsev condition

$$\mathcal{NU}(k) := \{m(x, x, \ldots, x, y) \approx x,\ m(x, x, \ldots, x, y, x) \approx x,\ \ldots,\ m(y, x, \ldots, x) \approx x\}$$

in which m is a k-ary operation symbol and there is one equation for each position of the "lone dissenter" y.

The study of near unanimity terms has been underway since at least the 1970's, when Baker and Pixley ([2]) proved the following celebrated result concerning satisfaction of this Mal'tsev condition.

Theorem 1.2.3 ([2], Theorem 2.1)**.** *Let $\mathcal{V}$ be a variety and $d \geq 2$ a natural number. The following are equivalent:*

1. *$\mathcal{V}$ has a $(d+1)$-ary near unanimity term*
2. *If $\mathbf{A}$ in $\mathcal{V}$ is a subalgebra of a direct product $\mathbf{P} = \mathbf{C}_1 \times \cdots \times \mathbf{C}_r$ where $r \geq d$ and $\mathbf{C}_i \in \mathcal{V}$ for each i, then $\mathbf{A}$ is uniquely determined by its d-fold coordinate projections. I.e. if $\mathbf{B}$ is another algebra in $\mathcal{V}$ with $\mathbf{B} \leq \mathbf{P}$ and the projection of B onto every choice of d factors $C_{i(1)}, \ldots, C_{i(d)}$ agrees with the projection of A onto these d factors, then $\mathbf{A} = \mathbf{B}$.*

[4] We will also occasionally refer to strong Mal'tsev conditions as Mal'tsev conditions, and may use the term *Mal'tsev condition* to mean *"strong or not strong" Mal'tsev condition.*

[5] An equation is called linear if the terms related each involve at most one function symbol.

3. *Given any algebra* $\mathbf{A} \in \mathcal{V}$*, if* r *congruences* $x \equiv a_i \bmod \theta_i$ $(1 \leq i \leq r$ *and* $r \geq d)$ *are solvable* d *at a time, then they are solvable simultaneously*[6]

4. *Given any algebra* $\mathbf{A}$ *in* $\mathcal{V}$ *and any partial function* $f : A^n \to A$ *for some* $n \geq 1$*, if every function* $f|_D$ *agrees with the restriction of some term operation* $t_D^{\mathbf{A}}$ *of* $\mathbf{A}$*, where* D *ranges over all the subsets of* A *on which* f *is defined with* d *or fewer elements, then* f *agrees with the restriction of some term operation* $t_f^{\mathbf{A}}$ *of* $\mathbf{A}$ *on the whole domain of* f

5. *Given any algebra* $\mathbf{A}$ *in* $\mathcal{V}$ *and any partial function* $f : A^n \to A$ *for some* $n \geq 1$*,* f *is the restriction of some term operation* $t_f^{\mathbf{A}}$ *of* $\mathbf{A}$ *to the domain of* f *if and only if every subset of* A^d *is closed under*[7] *the operation* f

Looking at the above theorem, we see that satisfying the strong Mal'tsev condition "Existence of a $(d+1)$-ary near unanimity term" gives insight into the existence of solutions of simultaneous congruences (item 3) as well as the ability to interpolate partial functions by term operations (items 4 and 5). We also see in item 2 that subalgebras of products can be recognized or distinguished simply by looking at the d-fold projections of those algebras. The import of this Mal'tsev condition is therefore immediately apparent from this theorem. Other nice properties also follow.

Further to the useful properties outlined above, it was established in [28] that if an algebra $\mathbf{A}$ satisfies the existence of a $(d+1)$-ary near unanimity term, then CSP($\mathbf{A}$) is tractable (see Definition 2.2.1). This provides an early example of the use of Mal'tsev conditions to establish tractability results in the field of constraint satisfaction, a programme of research which led recently to the celebrated Algebraic Dichotomy Theorem for Constraint Satisfaction Problems ([14, 44]). Problems in constraint satisfaction serve as vital motivation for many active areas of research, including the study of Mal'tsev Condition Satisfaction Problems which became the focus of this thesis. Results on algebras of "bounded width" (see Sections 5.3, 5.5, and 5.6 of the survey article [5] for an overview of these ideas) which extend the results found in [28] will be used later on to establish a polynomial-time decision procedure in Theorem 3.1.8. Near unanimity terms also serve as a special case of results outlined in Chapter 4 (Theorem 4.4.5) and Chapter 5 (Theorem 5.2.4).

Example 1.2.4. The strong Mal'tsev condition of having a k-ary near unanimity term (see Example 1.2.2) naturally suggests the following Mal'tsev condition. We say that $\mathbf{A}$ *has a near unanimity term* if $\mathbf{A}$ satisfies $\mathcal{NU}(k)$ for some $k \geq 3$. I.e. $\mathbf{A}$ has a k-ary near unanimity term for some $k \geq 3$. To see that this is a Mal'tsev condition (as defined in Definition 1.2.1) note that given a k-ary near unanimity term $m(x_1, \ldots, x_k)$

[6] Congruences are not directly used in this thesis. See [2] for more information on the property described in this item.

[7] Following [2, Section 2], we say that $\mathbf{S} \leq \mathbf{A}^k$ is *closed under* f if for any choice of n k-tuples in S the k-tuple obtained by applying f coordinate-wise to these tuples is also in S whenever this tuple is defined.

of $\mathbf{A}$ we may define $m^+(x_1, \ldots, x_{k+1}) := m(x_1, \ldots, x_k)$ and it is not difficult to see that m^+ is a $(k+1)$-ary near unanimity term of $\mathbf{A}$. Hence $\mathcal{NU}(k)$ implies $\mathcal{NU}(k+1)$ for each $k \geq 3$, so $\mathcal{NU} := (\mathcal{NU}(k))_{k\geq 3}$ is a Mal'tsev condition.

The following Mal'tsev conditions all make an appearance in Chapter 4. Motivation for considering each condition is discussed following each example.

Example 1.2.5. Fix $k \geq 2$. A *k-ary cyclic term* is a term $t(x_1, \ldots, x_k)$ which satisfies the equation $t(x_1, \ldots, x_k) \approx t(x_2, \ldots, x_k, x_1)$.

The strong Mal'tsev condition "existence of a k-ary cyclic term" is a compelling object of study because, similarly to the condition $\mathcal{NU}(k)$ outlined above, the condition is extremely simple to define and has very interesting structural consequences. This condition is explored in great detail in [4] and [3]. In particular, it is shown in [3, Theorem 4.1] that a finitely generated idempotent variety[8] has a cyclic term of some arity if and only if it satisfies a nontrivial Mal'tsev condition.

In this thesis, we introduce k-ary cyclic terms as a first example of a Mal'tsev condition whose satisfaction in a given finite algebra can be determined in polynomial-time[9] and for which term operations witnessing the satisfaction can also be obtained[10] in polynomial-time. This result is presented in Section 4.2 and is the first of many similar results for other Mal'tsev conditions which make up the content of Chapter 4. This line of research is inspired by the result of [29, Theorem 6] in which the same result is obtained for "Mal'tsev Terms". That result is reproduced in this thesis as Theorem 4.3.1 and used as a guiding example to understanding Theorem 4.4.5. We introduce Mal'tsev terms formally in the next example.

Example 1.2.6. A *Mal'tsev term* is a ternary term $p(x, y, z)$ which satisfies the equations $p(x, x, y) \approx y$, and $p(y, x, x) \approx y$.

As suggested by the name, Mal'tsev terms have a special place in universal algebra as the original Mal'tsev condition. Introduced by Anatoly Mal'tsev in [35], it was demonstrated in that paper that any given variety $\mathcal{V}$ satisfies the Mal'tsev condition of having a Mal'tsev term if and only if the variety $\mathcal{V}$ is "congruence permutable". While a discussion of congruences and the properties of congruence varieties eludes the scope of this thesis, we find it pertinent to mention this result since it provides an early example of the deep connection between structural algebraic properties and the satisfaction of Mal'tsev conditions.

The condition of having a Mal'tsev term has also played a critical role in the development of algorithms for constraint satisfaction problems, since it was demonstrated in [14] that if $\mathbf{A}$ has a Mal'tsev term, then CSP($\mathbf{A}$) is tractable. This result was

[8] A variety is said to be *idempotent* if each algebra in the variety is idempotent (see Definition 1.1.1).

[9] See Chapter 2.

[10] As operation tables or as circuits (see Definition 2.4.1).

naturally extended in [8] to include all algebras satisfying the Mal'tsev condition of having an "edge term" (defined in the next example). The satisfaction of this Mal'tsev condition also leads to tractability results for certain subclasses of the decision problem SMP($\mathbf{A}$)[11] as demonstrated by Mayr in [36]. Again, this pioneering result led to the extension given in [13] to the condition of having an edge term. Following this pattern, in this thesis we extend the result of [29, Theorem 6] which concerns Mal'tsev terms to include a much broader class of conditions which includes k-edge terms for fixed k.

Example 1.2.7. Fix $k \geq 2$. The strong Mal'tsev condition of *having a k-edge term* is given by the equations:

$$\begin{gathered}\{t(y,y,x,x,\dots,x) \approx x,\\ t(y,x,y,x,\dots,x) \approx x,\\ t(x,x,x,y,\dots,x) \approx x,\\ \vdots\\ t(x,x,x,\dots,y,x) \approx x,\\ t(x,x,x,\dots,x,y) \approx x\}\end{gathered}$$

where t is a $(k+1)$-ary operation symbol and all instances of "..." are the appropriate number of repeated x variables.

The condition of having a 2-edge term $\{t(y,y,x) \approx x, t(y,x,y) \approx x\}$ is therefore clearly equivalent to that of having a Mal'tsev term (defined in the previous example). It is also clearly the case that $\mathcal{NU}(k)$ implies the existence of a k-edge term, and hence this strong Mal'tsev condition can be thought of as a simultaneous generalization of both Mal'tsev and k-ary near unanimity terms. The Mal'tsev condition *existence of an edge term* is given by the sequence $(E(k))_{k\geq 2}$ where $E(k)$ is existence of a k-edge term. The argument that this is a Mal'tsev condition is similar to that in Example 1.2.4.

The condition of having an edge term is another example of a Mal'tsev condition with profound structural consequences for the models of this condition. Of particular importance is the result demonstrated in [8, Theorem 3.10] that having an edge term precisely characterizes those finite algebras with "few subpowers". Similarly to the case of having Mal'tsev terms outlined in [14], it is shown in [25] that CSP($\mathbf{A}$) is tractable if $\mathbf{A}$ satisfies the Mal'tsev condition of having an edge term. The technique in both examples is based on finding small generating sets for subpowers of $\mathbf{A}$ with particular special properties[12]. This is the same technique used in [36] and later in

[11] See Definition 2.2.3.

[12] See the referenced papers for more details.

[13] to show that SMP($\mathbf{A}$) is in NP[13] whenever $\mathbf{A}$ satisfies the condition of having a Mal'tsev term (in [36]) or more generally, an edge term (in [13]). This fact is then used to provide a nondeterministic polynomial-time algorithm for the idempotent Mal'tsev condition satisfaction problem for the condition of having a "minority term" (defined in the next example) in [29] and extended in this thesis to any condition of height < 1 (see Chapter 5).

Edge terms also make an appearance in Chapter 4 as a special case of Theorem 4.4.5. This result (Corollary 4.4.7) is then used in Chapter 5 to build parallelogram term operations (defined in Example 1.2.12) which are essential in the polynomial-time verifier of Theorem 5.2.4 (and subsequently in Theorem 6.1.4). The Mal'tsev condition "existence of an edge term" turns out to be equivalent to the property of "having a cube term" defined in Example 1.2.9 which is not a Mal'tsev condition according to Definition 1.2.1 but is rather a collection of Mal'tsev conditions. More details follow that example.

Example 1.2.8. The Mal'tsev condition of *having a minority term* is defined via

$$\text{Minority} := \{m(x,y,y) \approx x,\ m(y,x,y) \approx x,\ m(y,y,x) \approx x\}$$

.

The case of Minority Terms is particularly interesting for those examining the complexity of Mal'tsev condition satisfaction problems. Syntactically, it is very similar to the $\mathcal{NU}(3)$ condition outlined in Example 1.2.2 (here we choose the minority input rather than the nearly unanimous input) and also to the condition of having a Mal'tsev term outlined in Example 1.2.6. Freese and Valeriote (in [20]) demonstrated that the idempotent Mal'tsev condition satisfaction problems for both of these syntactically similar strong Mal'tsev conditions ($\mathcal{NU}(3)$ and Mal'tsev) are in the class P.[14] These results, together with alternative algorithms given in [24] for these and other syntactically close Mal'tsev conditions led to the belief that the Mal'tsev condition satisfaction problem for minority terms might also be polynomial-time solvable via a "local-global" type algorithm (i.e. of the kind developed in [24, Section 2]). In [29, Section 5] it is shown that this kind of approach will not work for minority terms but the problem is nevertheless in the complexity class NP. In this thesis we are able to extend this NP result to include all conditions of height < 1 (defined in Chapter 5) but it is still unknown whether the idempotent Mal'tsev condition satisfaction problem for minority terms is actually tractable.

Example 1.2.9. A strong Mal'tsev condition is called a *cube term condition* if it involves only a single operation symbol and the equations satisfy the following conditions:

[13] See Chapter 2.
[14] See Chapter 2.

- Every equation is of the form $t(x_1, \ldots, x_n) \approx x$ where x is some (fixed) variable and $x_i \in \{x, y\}$ for each $i = 1, \ldots, n$
- If we arrange the equations into a single matrix equation $t(M) \approx \overline{x}$, then each column of M contains at least one y. I.e. for each position $i \in \{1, \ldots, n\}$ of the inputs to t there is an equation $t(x_1, \ldots, x_n) \approx x$ in which x_i is y.

If $\mathbf{A}$ is an algebra which satisfies a particular cube term condition Σ, we call any term t of $\mathbf{A}$ which witnesses the satisfaction of Σ a *cube term* of $\mathbf{A}$. If the matrix M given by the equations of Σ is a $(k \times n)$-matrix, we call t a *k-cube term (of arity n).*

Although the property "having a cube term" is not formally a Mal'tsev condition (according to our Definition 1.2.1), the following theorem demonstrated in [8] shows that the existence of a cube term for $\mathbf{A}$ is equivalent to a Mal'tsev condition:[15]

Theorem 1.2.10 ([8], Theorem 2.12)**.** *Let $\mathbf{A}$ be an algebra and $k \geq 2$. Then $\mathbf{A}$ has a k-cube term (of some arity) if and only if $\mathbf{A}$ has a k-edge term (of arity $(k+1)$).*

The theorem reveals the nature of the k-edge Mal'tsev condition as a generic cube term condition. The definition of a k-cube term is in itself already very general. In Section 5.2 we outline in detail how any nontrivial strong Mal'tsev condition of height < 1 implies the existence of a k-cube term (for some k depending on the given condition). This highlights the generality of the definition given in Example 1.2.9. In particular, any result on cube terms (for example the result of Theorem 5.2.4) also applies to the following particular examples of cube term conditions:

- Near unanimity terms (see Example 1.2.2)
- Mal'tsev terms (see Example 1.2.6)
- Edge terms (see Example 1.2.7)
- Minority terms (see Example 1.2.8)

Example 1.2.11. All of the examples seen above are relatively simple strong Mal'tsev conditions since they only contain a single operation symbol. Of course we are often interested also in Mal'tsev conditions which relate many different operation symbols. Fix $n \geq 1$. The strong Mal'tsev condition *existence of a sequence of n Hagemann-Mitschke terms* is given by the following equations:

$$\{p_1(x, y, y) \approx x,\ p_i(x, x, y) \approx p_{i+1}(x, y, y) \text{ for } 1 \leq i \leq n-1,\ p_n(x, x, y) \approx y\}$$

[15] The definition of k-cube term used in this thesis corresponds to the definition of δ-special cube terms in [8]. The two are shown to be equivalent in that paper ([8, Theorem 2.12]).

It was shown in 1973 [21] that the algebra $\mathbf{A}$ has a sequence of n Hagemann-Mitschke terms if and only if the variety generated by $\mathbf{A}$ is congruence $(n+1)$-permutable. In [22] and [31] the Mal'tsev condition of "having a sequence of Hagemann-Mitschke terms" (i.e. "being n-permutable for some n") is explored in some detail and this programme of study is explored further in [42]. Once again, we elect to omit any detailed exploration of these interesting results and jump straight to the complexity theoretic questions at hand. In particular, it has already been shown that the idempotent Mal'tsev condition satisfaction problem is in P for the conditions "having a sequence of n Hagemann-Mitschke terms" (proven in [42]) and "having a sequence of Hagemann-Mitschke terms" (proven in [20]). In Chapter 4 of this thesis, we are interested in the related search problem of finding appropriate witnesses (term operations of $\mathbf{A}$) for satisfaction of various Mal'tsev conditions. In particular, for the case of the strong Mal'tsev condition "having a sequence of n Hagemann-Mitschke terms", we find that witnesses for the satisfaction of this condition can indeed be obtained in polynomial-time. We also conjecture (Conjecture 4.4.11) that this result may be extended to include all of the "Path Mal'tsev Conditions" outlined in [30] but as yet that question remains open.

The last Mal'tsev condition to be introduced in this section finds a use in the proof of Theorem 5.2.4.

Example 1.2.12. Fix $m, n \geq 1$ and set $k := m+n$. The Mal'tsev condition *"existence of an (m,n)-parallelogram term"* is given by the m equations:

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

together with the n equations:

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

all involving the $(k+3)$-ary operation symbol P.

In [32, Theorem 3.5] the existence of an (m,n)-parallelogram term is shown to be equivalent to the existence of an $(m+n)$-edge term (and hence also to the existence of an $(m+n)$-cube term). In [13, Theorem 4.13] the satisfaction of this Mal'tsev

condition by an algebra $\mathbf{A}$ is used to show that SMP($\mathbf{A}$) is in NP. Definitions of SMP and NP can be found in Chapter 2 and the proof of Theorem 5.2.4 involves reproducing key sections of [13] to see that the same techniques can be used to solve $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ when Σ implies the existence of an (m, n)-parallelogram term.

This chapter has introduced all the relevant notions from universal algebra and outlined all of the Mal'tsev conditions which will be considered in this thesis. In the next chapter we introduce the relevant notions from complexity theory and outline the class of decision problems which are the focus of the thesis, namely Mal'tsev condition satisfaction problems.

Chapter 2

On the Complexity of Mal'tsev Condition Satisfaction Problems

In this chapter we introduce the basic notions of complexity theory necessary to understand the results and proofs found in the remainder of this thesis. In particular, we introduce the problems which we are concerned with solving and the relevant complexity-theoretic definitions to describe how difficult these problems are. For the purposes of this thesis, we will not concern ourselves with the details of any particular model of computation, nor do we find it convenient to examine the precise definition of Turing machines and the relative complexities of single vs. multitape machines and related problems. We regard these finer details of the theory of computation as an interesting but ultimately unnecessary distraction.

The aim of this chapter then, is to provide a basic description of the main complexity classes referenced in the results of this thesis. The emphasis when describing the time taken to complete a specific decision procedure is on how the time changes as a function of the input size, rather than any concrete description of how much time a computation takes. For this reason we find it sufficient to present algorithms in natural language as a sequence of instructions. We analyze how many times a specific instruction will have to be implemented and how long each implementation will take – viewing both of these as functions in the size of the input. Of course, the actual time taken to implement any algorithm outlined in this text will depend heavily on the machine which runs the computation which justifies our decision to analyze these algorithms independently of any particular implementation. We do however provide specific runtime estimates for the algorithms provided in our proofs when such an analysis is possible, in the standard "Big O" notation defined in Definition 2.1.2. Where these are provided, we make no claim that our decision procedures are optimal, merely that these upper bounds apply for the algorithms defined. For a more thorough introduction to complexity theory the reader is directed to [40].

2.1 O, P, NP and EXPTIME

Later in this thesis we will be interested in analyzing the complexity of certain decision problems. In order to do so, we first need to define what we mean by certain complexity theoretic concepts. We prefer not to take too technical a dive into the intricacies of complexity theory. We begin with an informal definition of "Decision Problems".

Definition 2.1.1. A *decision problem* is given by an infinite set of allowed inputs each of which is either a YES instance of the problem or a NO instance. A *decision procedure* for a decision problem is an algorithm implementable by a Turing machine which correctly decides whether a given input is a YES instance or a NO instance of the problem.

It is important to remark that a given decision problem may have no decision procedure according to our definition (in which case the problem is called *undecidable*).

For decidable problems, we are interested in knowing the theoretical limits on how 'fast' a decision procedure could be. As noted in the introduction to this chapter, our concern here is not specifically how long a given Turing machine would take to answer the YES/NO question but rather on how the time taken to implement the decision procedure changes in proportion to the 'size' of the input (for some reasonable notion of 'size' which may depend on the problem in hand). The results of this thesis are all essentially of the nature: "For decision problem D, there is a decision procedure A for which increasing the size of the input does not lead to a corresponding exponential increase in the time taken to run procedure A"[1].

In order to make this idea more precise, we introduce three classes of decision problems called "complexity classes" and the results of the thesis are then precisely stated as "Decision problem D lies in complexity class C". Before we do so, we need the following tool of complexity theory.

Definition 2.1.2. Let $f, g : \mathbb{N} \to \mathbb{R}$ be functions. We say that $f(n)$ *is* $O(g(n))$ if there exist positive integers c and N such that for all $n > N$ we have $f(n) \leq cg(n)$.

The expression $f(n)$ is $O(g(n))$ therefore means that 'eventually' $f(n)$ is at most $cg(n)$. The function g is thought of as an upper bound of f in this case. We also occasionally write $f(n) = O(g(n))$, remembering that "=" in this usage is not symmetric. When $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$ we will also write (for example) $f_1(n) + f_2(n) = O(g_1(n)) + O(g_2(n)) = O(h(n))$ where $h(n)$ is any function for which $g_1(n)$ and $g_2(n)$ are both $O(h(n))$. This expresses the fact that "eventually" the combined value $f_1(n) + f_2(n)$ is at most $ch(n)$ for some c. Once again, it is important to note that this use of equality is not symmetric and the usual laws of arithmetic do not apply. For example, it does not follow from the previous expression that

[1] In the case of Theorems 3.2.1 and 5.2.4 this description is only valid if $\mathsf{NP} \neq \mathsf{EXPTIME}$.

$O(h(n)) - O(g_1(n)) = O(g_2(n))$. Indeed, we ascribe no meaning to the expression $O(h(n)) - O(g_1(n))$.

We are now ready to define two important complexity classes. Examples of problems in each class will be given in Section 2.3.

Definition 2.1.3. Let D be a decision problem.

- We say that D *is tractable* or *polynomial-time solvable* if there is a decision procedure A for D and a Turing machine implementing A whose runtime[2] $f(n)$ is $O(n^k)$ for some $k \in \mathbb{N}$.
- The complexity class P is defined to be the class of all decision problems which are tractable.
- We say that D is *exponential-time solvable* if there is a decision procedure A for D and a Turing machine implementing A whose runtime $f(n)$ is $O(2^{p(n)})$ for some polynomial $p(n)$.
- The complexity class $\mathsf{EXPTIME}$ is defined to be the class of all decision problems which are solvable in exponential time.

The third complexity class that we are interested in has a slightly different definition:

Definition 2.1.4. Let D be a decision problem.

- A *verifier* V for D is an algorithm implementable by a Turing machine with the following properties:
  - For any YES instance Y of D there is a corresponding *certificate* $C(Y)$ such that when V is run on input $(Y, C(Y))$ the output is YES
  - For any NO instance N of D and any string C the algorithm V gives output NO on input (N, C).
- We say that V is a *polynomial-time verifier* for D if there is a Turing machine implementing V whose runtime $f(n)$ (where n is the size of the instance of D) is $O(n^k)$.
- The complexity class NP is the class of all decision problems for which there is a polynomial-time verifier.

[2]The runtime of a Turing machine is defined as the function $f : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ where $f(n)$ is the maximum number of steps that the Turing machine takes before halting on any input of size n.

It is immediately clear from Definitions 2.1.4 and 2.1.3 that $\mathsf{P} \subseteq \mathsf{NP}$ (for a verifier, just use the polynomial-time algorithm which solves D). Less immediate (see [40, Section 8.2] for a proof) is the inclusion $\mathsf{NP} \subseteq \mathsf{EXPTIME}$. Later in that book ([40, Corollary 9.13]) it is demonstrated that $\mathsf{P} \neq \mathsf{EXPTIME}$ which demonstrates that at least one of the inclusions $\mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{EXPTIME}$ is proper. Many researchers in complexity theory (including the author of this thesis) believe both inclusions to be proper but as yet this remains an open problem. Occasionally in the thesis we may allow ourselves the liberty of discussing complexity theoretic results as if it were the case that $\mathsf{P} \subsetneq \mathsf{NP} \subsetneq \mathsf{EXPTIME}$ and we take the opportunity now (and as necessary later in the text) to remind the reader that it may indeed be the case that $\mathsf{P} = \mathsf{NP}$ or $\mathsf{NP} = \mathsf{EXPTIME}$. If $\mathsf{P} = \mathsf{NP}$ then Theorems 3.2.1 and 5.2.4 become tractability results, whereas if $\mathsf{NP} = \mathsf{EXPTIME}$ then those theorems are rendered trivial as $\mathsf{EXPTIME}$ results are already well-known for the conditions therein.

Definition 2.1.5. Let D be a decision problem.

- We say that D is *NP-hard* if given any NP-problem C there is an algorithm $A_{C\to D}$ implementable in time $O(n^k)$ for some $k \in \mathbb{N}$ such that given an instance I^C of C the algorithm $A_{C\to D}$ correctly produces an instance I^D of D which satisfies

$$I^D \text{ is a YES instance of } D \iff I^C \text{ is a YES instance of } C$$

- D is *NP-complete* if D is in NP and D is NP-hard.

- We say that D is *$\mathsf{EXPTIME}$-hard* if given any $\mathsf{EXPTIME}$-problem C there is an algorithm $A_{C\to D}$ implementable in time $O(n^k)$ for some $k \in \mathbb{N}$ such that given an instance I^C of C the algorithm $A_{C\to D}$ correctly produces an instance I^D of D which satisfies

$$I^D \text{ is a YES instance of } D \iff I^C \text{ is a YES instance of } C$$

- D is *$\mathsf{EXPTIME}$-complete* if D is in $\mathsf{EXPTIME}$ and D is $\mathsf{EXPTIME}$-hard.

The notion of K-completeness is generally thought of as capturing the "hardest" problems in the class K. While we do not establish any hardness results in this thesis, we will make reference to certain problems which are proven elsewhere to be $\mathsf{EXPTIME}$-complete and we will occasionally make reference in our discussions to problems which are NP-complete. The definitions introduced in this chapter represent only a tiny region of what is popularly referred to as the *complexity zoo*. For a glimpse at all the other beasts in this unique menagerie see the Complexity Zoo website maintained by the University of Waterloo at https://complexityzoo.uwaterloo.ca/Complexity_Zoo.

Having introduced decision problems and some of the complexity classes where they live, we are now ready to introduce a specific class of decision problems whose

importance to computer science and connection to universal algebra has invigorated the study of Mal'tsev conditions over the last thirty years. We also introduce a class of decision problems with classical origins in the theory of groups and modern applications in the fields of computational group theory and machine learning. Universal algebraic results concerning the problems in both of these classes have promoted and supported research into the class of decision problems which are the main focus of this thesis, which will be introduced in Section 2.3.

2.2 CSP and SMP

The constraint satisfaction problem (CSP) is a general yet structured framework in which to discuss many and various computational problems. Essentially, an instance of the constraint satisfaction problem is a collection of variables which must be assigned values from a given domain subject to various "constraints"- collections of variables must belong to certain relations over the domain. Clearly this description is broad enough to fit many problems of interest to computer scientists as well as many problems of practical importance in everyday applications. For example it may be that we wish to colour the regions of a map in such a way that no two adjacent regions have the same colour. Or to allot class times for several different courses such that no two classes which both appear on one syllabus are scheduled to occur at the same time. The decision version of the CSP is simply the question of deciding whether or not such an assignment of the variables is possible. It is well-known that this problem is NP-complete in general. However by limiting the allowed constraint relations one may obtain tractable versions of the problem.

This study of so-called nonuniform CSPs (subclasses of CSP where the allowed constraint relations are restricted) was initiated when Schaefer showed in 1978 [38] that if the domain is a two-element set then restricting the allowable constraint relations either gives an NP-complete class or a tractable class. An earlier 1975 result by Ladner [33] showed that if $\mathsf{P} \neq \mathsf{NP}$ (which is widely believed to be the case) then there are problems which lie between these two classes, called *NP-intermediate problems*. Schaefer's dichotomy theorem viewed from this perspective says that nonuniform CSPs over two-element domains never fall into this NP-intermediate region. They are either hard (NP-complete) or easy (in P).

It was conjectured in 1993 [17] that this result was also true for nonuniform constraint satisfaction problems over larger domains. Much focus was devoted to extending Schaefer's dichotomy theorem over the next decades and in 2017 it was finally proved independently by Bulatov [14] and Zhuk [44] that restricting the allowed constraints to a prescribed finite set Γ of relations (called a *constraint language*) yields either an NP-complete decision problem (as in the general case) or a tractable problem.

The dividing line between NP-complete CSPs and those which are tractable is shown in both [14] and [44] to be the satisfaction of a particular Mal'tsev condition

in a related algebra, as originally conjectured in [12]. Indeed many of the results establishing tractable subclasses are dependent upon satisfaction of various Mal'tsev conditions (see [5] for a survey of results in this area). This connection between Mal'tsev conditions and computational decision problems provides one of the most immediate motivations for the study of Mal'tsev conditions, leading inevitably to the consideration of the "meta-question" of Mal'tsev condition satisfaction (see Section 2.3). Before we ask ourselves the meta-question, we formalize the definition of nonuniform constraint satisfaction problems in the manner most convenient for subsequent use.

Definition 2.2.1. Let $\mathbf{A}$ be a finite algebra. We define the decision problem CSP($\mathbf{A}$) *(the constraint satisfaction problem over* $\mathbf{A}$*)* to be the problem whose instances are given by:

- INPUT:
  - A finite set V of *variables*
  - The set A (the universe of $\mathbf{A}$) is assumed to be part of the input and is called the *domain* of the instance
  - A finite set $\mathcal{C}$ of *constraints*, each of which is a pair $(\overline{s}, C)$ where $\overline{s}$ is a tuple of variables from V^n (for some $n \in \mathbb{N}$) and C is a subuniverse of $\mathbf{A}^n$
- QUESTION: Is there a function (called a *solution of the instance*) $f : V \to A$ such that for every constraint $(\overline{s}, C)$ we have $f(\overline{s}) \in C$ (where $f(\overline{s})$ is the tuple obtained by applying f to $\overline{s}$ coordinate-wise)?

As mentioned in the discussion before Definition 2.2.1, Bulatov [14] and Zhuk [44] have recently shown that the decision problem CSP($\mathbf{A}$) is either NP-complete or tractable. The problem CSP($\mathbf{A}$) is a broad enough class that in some cases we can frame instances of the Mal'tsev condition satisfaction problem (see Section 2.3) as instances of CSP($\mathbf{A}$) for the appropriate choice of $\mathbf{A}$. This technique is used in this thesis to establish the proof of Theorem 3.1.8. The algebra $\mathbf{A}$ involved in that proof certainly satisfies the (equivalent) tractability conditions outlined in [14, Theorem 2] and [44, Theorem 1.4] and so either of these results are sufficient to establish the tractability of the problem under consideration in Theorem 3.1.8. However in that particular case, the full generality of [14, Theorem 2] and [44, Theorem 1.4] is unnecessary and we instead make reference to the following earlier result of [28].

Theorem 2.2.2 (See [28], Corollary 3.6)**.** *Let* $\mathbf{A}$ *be an algebra which satisfies the Mal'tsev condition* $\mathcal{NU}$*. Then CSP(*$\mathbf{A}$*) is tractable.*

As previously mentioned, Theorem 2.2.2 is one example among many of the satisfaction of Mal'tsev conditions being used to establish tractability for restrictions of the CSP. Results of this kind not only serve to motivate consideration of the Mal'tsev

condition satisfaction problem outlined in the next section but they can then also be used (as in Theorem 3.1.8) to establish tractability results within that arena - an aesthetically pleasing observation to note. The next definition introduces another class of decision problems which serve a similar role as both inspiration for study and instrument of proof.

Definition 2.2.3. Let $\mathbf{A}$ be an algebra. The decision problem SMP($\mathbf{A}$) *(the subpower membership problem for* $\mathbf{A}$*)* is given by:

- INPUT: Finitely many tuples $\overline{a}, \overline{b}_1, \ldots, \overline{b}_k \in A^n$ (for some $n \in \mathbb{N}$)
- QUESTION: Is $\overline{a}$ in the subpower of $\mathbf{A}$ generated by $\overline{b}_1, \ldots, \overline{b}_k$ (i.e. $\overline{a} \in \langle \overline{b}_1, \ldots, \overline{b}_k \rangle_{\mathbf{A}^n}$?)

The subpower membership problem for $\mathbf{A}$ is a generalization of a question familiar to computational group theorists known as the subgroup membership problem, in which $\mathbf{A}$ is a group and $n = 1$. In the general case described here, we see that the input has size $O(n(k+1))$. The result of [20, Proposition 6.1] (alternatively see Proposition 4.1.1) therefore implies that this problem is in the class EXPTIME when $\mathbf{A}$ is a finite algebra. Of course placing further restrictions on the algebra $\mathbf{A}$ may give rise to decision problems in NP or even in P.

Notably from our perspective, Mayr shows in [36] that SMP($\mathbf{A}$) is in the class NP when $\mathbf{A}$ is a finite algebra which satisfies the Mal'tsev condition of having a Mal'tsev term (see Example 1.2.6). That result uses what Mayr calls "canonical representations" for subpowers of $\mathbf{A}$- special generating sets of small size given which membership can be checked in polynomial-time. These representations are based on earlier work found in [1] and [11]. In [1] Aichinger uses these ideas to bound the number of distinct finite algebras which satisfy the Mal'tsev condition of having a Mal'tsev term, whereas in [11] representations are used to solve CSP($\mathbf{A}$) when $\mathbf{A}$ has a Mal'tsev term.

The construction of these representations is extended to include any algebra satisfying a cube term condition in [8] where it is shown that such algebras also give rise to tractable subclasses of CSP. This construction is then used in [13] to demonstrate that SMP($\mathbf{A}$) is in NP whenever $\mathbf{A}$ is a finite algebra with a cube term and indeed in the class P if $\mathbf{A}$ satisfies the additional property of generating a residually small variety (not defined in this thesis).

As remarked before Definition 2.2.3 these complexity results for SMP($\mathbf{A}$) can also be used to derive corresponding results for Mal'tsev condition satisfaction problems. In [29] it is Mayr's result on SMP($\mathbf{A}$) for algebras with a Mal'tsev term which is used to show that $\mathrm{Sat}^{\mathrm{Id}}_{\mathrm{Minority}}$ is in the class NP.[3] Similarly, we use the NP result of [13] to derive Theorem 5.2.4 of Chapter 5. Much of the analysis of [13, Section 2] is reproduced in Section 5.2 because we will be interested on the complexity of those

[3] See Definition 2.3.1.

algorithms as a function of $||\mathbf{A}||$ where $\mathbf{A}$ is an instance of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ whereas in the analysis of [13] the background algebra $\mathbf{A}$ is a fixed constant for any instance of SMP($\mathbf{A}$).

2.3 MCSP

We are now ready to introduce the class of decision problems about which this thesis is concerned.

Definition 2.3.1. Let Σ be a Mal'tsev condition. The *Mal'tsev condition satisfaction problem (MCSP) for* Σ (or the Σ-*satisfaction problem*) is the following decision problem:

- INPUT: A finite algebra $\mathbf{A}$
- QUESTION: Does $\mathbf{A}$ satisfy Σ?

The *idempotent Mal'tsev condition satisfaction problem for* Σ is the related problem:

- INPUT: A finite idempotent algebra $\mathbf{A}$
- QUESTION: Does $\mathbf{A}$ satisfy Σ?

We denote the Σ-satisfaction problem by Sat_{Σ} and the idempotent Σ-satisfaction problem by $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$.

Researchers of the CSP will recognize this problem as the "meta-question" of constraint satisfaction. Chen and Larose in [16] considered a closely-related problem in which the input is a relational structure and the question is whether the algebra of polymorphisms satisfies the condition Σ. We now briefly survey current results concerning the algebraic version of the problem defined here.

We begin by listing (with references) Mal'tsev conditions whose idempotent satisfaction problem has been proven tractable. For definitions of those conditions not defined in Section 1.2 see the references listed for each result. For the most part we have elected to name each condition as the existence of certain terms. In some instances universal algebraists will perhaps be more familiar with the names of equivalent conditions (for finite algebras) listed in parentheses.

Theorem 2.3.2. *Sat^{Id}_{Σ} is in P whenever Σ is one of the following conditions:*

1. *Existence of a sequence of Day terms [20, Theorem 6.2] (congruence modularity)*
2. *Existence of a sequence of Jónsson terms [20, Theorem 6.2] (congruence distributivity)*
3. *Existence of a sequence of Hobby-McKenzie Terms [20, Theorem 6.2] (congruence join-semidistributivity)*

4. *Existence of ternary and quaternary weak near unanimity operations* $t(x, y, z)$ *and* $q(w, x, y, z)$ *satisfying* $t(x, y, y) \approx q(x, y, y, y)$ *[20, Theorem 6.2] (congruence meet-semidistributivity)*

5. *Existence of a Mal'tsev term [20, Theorem 6.2] (congruence permutability)*

6. *Existence of a sequence of Hagemann-Mitschke terms [20, Theorem 6.2] (congruence* n*-permutability for some* n*)*

7. *Existence of a majority term [20, Theorem 6.2]*

8. *Any condition satisfying the downward column condition [24, Theorem 2.6], including:*

   (i) *For fixed* $n > 2$*, existence of an* n*-ary near unanimity term [24, Corollary 2.7]*

   (ii) *For fixed* $k > 1$*, existence of a* k*-ary edge term [24, Corollary 2.7]*

9. *Existence of a Pixley term [24, Lemma 2.8] (generating an arithmetic variety)*

10. *For fixed* $n \geq 1$*, existence of a sequence of* n *Hagemann-Mitschke terms [42, Corollary 2.4] (congruence* $(n + 1)$*-permutability)*

11. *For fixed* $n > 1$*, existence of a sequence of* n *Jónsson terms [30, Corollary 8]*

12. *For fixed* $n > 1$*, existence of a sequence of* n *Gumm terms [30, Corollary 8]*

Conditions (1)-(7) of the above theorem were shown to be tractable in [20] based on the congruence properties listed in parentheses and with the help of technical results in "tame congruence theory". In particular, the algorithms involved are all based on generating subpowers of $\mathbf{A}$ for some fixed small powers. Conditions (8)-(12) are shown to be tractable in [24, 42, 30] by checking whether there are terms which satisfy the given equations (or related equations) on subsets of some small size (fixed for each condition) and then using technical results to conclude that there are terms which satisfy the equations globally. Again, this amounts in practice to generating small subpowers of the given algebra $\mathbf{A}$.

It is notable that each of these conditions is linear (see Definition 1.2.1). For linear conditions like many of those listed in the above result, compositions of terms which satisfy the equations locally (i.e. on certain small subsets) in a finite idempotent algebra can be shown to satisfy the given equations on slightly larger subsets in an inductive manner. It is not clear how well results like these can extend when the given equations are nonlinear. The next theorem shows that there are some strong Mal'tsev conditions for which the idempotent MCSP is not tractable. Notably the condition below is nonlinear.

Theorem 2.3.3. *Sat_Σ^{Id} is EXPTIME-complete when Σ is the Mal'tsev condition "existence of a semilattice term" [19, Theorem 4.5].*

As remarked above, existence of a semilattice term is a nonlinear Mal'tsev condition (see [41] for a proof). In light of the preceding two theorems a conjecture immediately presents itself: testing for the satisfaction of a linear strong Mal'tsev condition in an idempotent algebra is tractable, while testing for the satisfaction of a nonlinear strong Mal'tsev condition is EXPTIME-complete. As we shall see in Chapter 3, this conjecture is false. Corollary 3.2.3 in particular gives an example of a nonlinear (see again [41]) Mal'tsev condition whose idempotent satisfaction problem is an NP problem. Whether there are linear strong Mal'tsev conditions whose idempotent satisfaction problem requires superpolynomial-time is yet to be established (even under the assumption $\mathsf{P} \neq \mathsf{NP}$).

There are however several examples of linear and nonlinear Mal'tsev conditions whose satisfaction problem is EXPTIME-complete in general. Existing results are summarized in the next theorem.

Theorem 2.3.4. *Sat_Σ is EXPTIME-complete for the following conditions:*

1. *Existence of a semilattice term [20, Corollary 9.3]*
2. *Existence of a Taylor term [20, Corollary 9.3] (existence of a Siggers term [39])*
3. *Existence of ternary and quaternary weak near unanimity operations $t(x, y, z)$ and $q(w, x, y, z)$ satisfying $t(x, y, y) \approx q(x, y, y, y)$ [20, Corollary 9.3]*
4. *Omitting types 1 and 5 [20, Corollary 9.3]*
5. *Existence of a sequence of Hobby-McKenzie terms [20, Corollary 9.3]*
6. *Existence of a sequence of Day terms [20, Corollary 9.3]*
7. *Existence of a sequence of Jónsson terms [20, Corollary 9.3]*
8. *For fixed $n > 3$, existence of a sequence of n Jónsson terms [20, Corollary 9.3]*
9. *For fixed $n > 2$, existence of a sequence of n Hagemann-Mitschke terms [24, Corollary 3.6] (congruence n-permutability)*
10. *Existence of a sequence of Hagemann-Mitschke terms [24, Corollary 3.6]*
11. *Omitting types 1, 2, 4 and 5 [24, Corollary 3.6]*
12. *For fixed $n > 1$, existence of an idempotent cyclic term of arity n [23, Corollary 5.1.9]*
13. *For fixed $n > 1$, an n-ary weak near unanimity term [23, Corollary 5.1.9]*

In this thesis we will not add to the list of problems known to be EXPTIME-complete. We return briefly now to the idempotent Σ-satisfaction problem and the somewhat mysterious condition of minority terms. Recall from Examples 1.2.2 1.2.6, & 1.2.8 that the conditions $\mathcal{NU}(3)$, existence of a Mal'tsev term, and existence of a minority term are syntactically quite similar. It is interesting to note that the Mal'tsev condition Minority of Example 1.2.8 is missing from Theorem 2.3.2. In a recent paper of Kazda, Opršal, Valeriote and Zhuk [29] it is shown that techniques involving the local-global type argument described in the discussion following Theorem 2.3.2 may turn out to be ineffective in establishing tractability of the condition Minority. In particular, the authors of that paper provide a family of algebras of increasing size which each have term operations satisfying the minority equations on subsets of some size proportional to the size of the algebra but which do not have global minority operations.

This could suggest some initial evidence that the Mal'tsev condition Minority may give rise to an EXPTIME-complete MCSP. However, the authors of [29] rule that out[4] in the following theorem:

Theorem 2.3.5 ([29], Theorem 17)**.** $Sat^{Id}_{Minority} \in NP$.

The complexity of the idempotent Minority-satisfaction problem is still not known to be either tractable or NP-complete. It seems unlikely to the author of this thesis that $\mathrm{Sat}^{\mathrm{Id}}_{\mathrm{Minority}}$ is a genuine NP-intermediate problem but more research is needed to determine the precise complexity of this MCSP. In this thesis, we are at least able to extend the result of [29, Theorem 17]. In Theorem 5.2.4 we replace "Minority" with any Mal'tsev condition of height < 1 (see Definition 5.1.1) thus extending this result to a broad class of decision problems. As noted in [29], the complexity of Σ-satisfaction problems for such conditions seems to be closely linked with subpower membership problems over algebras which satisfy related conditions. See the discussion in Chapter 5 for more detailed analysis.

In the next chapter we begin to extend the boundaries of current knowledge by proving tractability and NP results for some conditions not yet established in the field.

2.4 Circuits

Let Σ be a Mal'tsev condition. Aside from the decision problems Sat_Σ and $\mathrm{Sat}^{\mathrm{Id}}_\Sigma$ we will also concern ourselves in this thesis with the related search problems:

- INPUT: A finite (idempotent) algebra $\mathbf{A}$.
- GOAL: Obtain operation tables for term operations of $\mathbf{A}$ which witness that $\mathbf{A}$ satisfies the Mal'tsev condition Σ if such terms exist.

[4] Once again assuming NP $\neq$ EXPTIME.

Clearly this is a "harder" problem than the decision problems Sat_Σ and $\mathrm{Sat}^{\mathrm{Id}}_\Sigma$ since in obtaining operation tables witnessing the satisfaction we also answer the question of whether the Mal'tsev condition is satisfied. One can easily imagine contexts in which an algorithm implementing the search problem would be of greater practical use than one which merely answers the decision problem. For example, in [11] an algorithm is given for deciding CSPs over Mal'tsev templates (i.e. instances of $\mathrm{CSP}(\mathbf{A})$ where $\mathbf{A}$ is an algebra with a Mal'tsev term) in which a Mal'tsev term operation of $\mathbf{A}$ is explicitly used to decide whether a solution exists. Hence in order to implement the algorithm it is necessary first to have a Mal'tsev term operation "in hand".

Similarly, the result of [13, Theorem 4.13] provides a polynomial-time verifier for $\mathrm{SMP}(\mathbf{A})$ whenever $\mathbf{A}$ is a finite algebra with a parallelogram term and to implement the algorithm requires explicitly evaluating the parallelogram term on particular inputs. The proof of Theorem 5.2.4 also uses the same verifier and hence provides another example when knowing that such a term operation exists is insufficient without being able to evaluate the term operation.

The usual presentation for an operation f on a finite set A is as a table listing the value $f(\overline{a})$ at each input tuple $\overline{a}$. *Circuits* provide an alternative representation of term operations which sometimes prove more effective in computations. We introduce the definition here and provide a reference for readers interested in further exploring the computational advantages.

Definition 2.4.1. Let $\mathbf{A} = (A, \mathcal{F})$ be an algebra, $n \geq 1$ and $t(x_1, \ldots, x_n)$ an n-ary term operation of $\mathbf{A}$. An *n-ary circuit C in the language of $\mathbf{A}$* is a finite directed, acyclic graph satisfying the following conditions:

- Each node of the circuit is designated as either an *input* or a *gate* and there are precisely n input nodes which are linearly ordered.
- Every input is a source (in-degree 0). The out-degree of an input may be any nonnegative integer.
- Every gate is labelled by an operation symbol f of a basic operation of $\mathbf{A}$. If the operation symbol f has arity k, then the in-degree of a gate labelled f is k and the k in-edges of the gate are linearly ordered. The out-degree of a gate may be any nonnegative integer.
- Some nodes are designated as *outputs* of the circuit and for convenience these are also assumed to be linearly ordered.

We define the *value of the n-ary circuit C on the input $a_1, \ldots, a_n \in A$* by first inductively defining the *value of a node of the circuit C on the input $a_1, \ldots, a_n$*:

- For $i = 1, \ldots, n$ the *value of the i-th input node on the input $a_1, \ldots, a_n$* is a_i.

- The *value* of a gate V labelled with the k-ary operation symbol f *on the input* $a_1, \ldots, a_n$ is $f^{\mathbf{A}}(v_1, \ldots, v_k)$ where v_i is the value of the i-th in-edge of the gate V on the input $a_1, \ldots, a_n$.
- The *value of* C *on the input* $a_1, \ldots, a_n$ is the value of the output nodes of C on the input $a_1, \ldots, a_n$.

We say that the n-ary circuit C is a *circuit for* $t(x_1, \ldots, x_n)$ if C has precisely one output and for any input $a_1, \ldots, a_n \in A$ we have that the value of C on $a_1, \ldots, a_n$ is $t(a_1, \ldots, a_n)$.

The *size* of the circuit C is defined to be the number of nodes of C and it should be clear that given operation tables for the basic operations of $\mathbf{A}$, the value of the circuit C on a given input $a_1, \ldots, a_n$ can be determined in time $O(R|C|)$ where R is the maximum in-degree of any node of C.

It should be clear from the above definition that circuits provide an alternative way to represent one or more term operations of $\mathbf{A}$. A simple example is given in Figure 2.1. A more elaborate example is found in Chapter 5 (Figure 5.1).

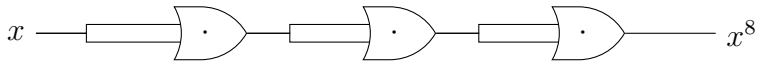

Figure 2.1: A Circuit in the Language of Groups for the Term $x^8 := x \cdot (x \cdot (x \cdot (x \cdot (x \cdot (x \cdot (x \cdot x))))))$

It is clear that any term operation of an algebra can be represented by a circuit since every term operation is built from the basic operations by generalized composition. Note that the circuit in Figure 2.1 has four nodes[5], whereas the circuit built based on the composition tree for the term x^8 would have eight input nodes and seven instances of the $\cdot$ gate.

Indeed in [26, Example 2.1] an example is provided which demonstrates that there really is a computational advantage to representing terms by circuits as compared to operation tables (assuming $\mathsf{P} \neq \mathsf{NP}$). For an exploration of the computational benefits of representing term operations by circuits the interested reader is directed to [26]. The benefits can also be seen within the pages of this thesis. In the proof of Theorem 5.2.4 we make use of circuits for term operations of an algebra $\mathbf{A}$ which can be built and evaluated on a single input in polynomial-time with respect to $||\mathbf{A}||$ but for which an operation table could not be effectively calculated (or even printed) in less than exponential time.

[5]The "node" labelled x^8 is included in the illustration but does not formally make up part of the circuit.

Chapter 3

Polynomial-Time and NP Results Using Near Unanimity Terms

Let's write a Haiku
To achieve that noble goal
Unanimity

In this chapter we examine how the interpolation results of Baker and Pixley [2] can be used to provide an NP certificate for the satisfaction of strong Mal'tsev conditions which imply the existence of near unanimity terms. This result implies the existence of nonlinear Mal'tsev conditions whose satisfaction problem is not EXPTIME-complete (assuming EXPTIME $\neq$ NP). In the case of linear strong Mal'tsev conditions which imply near unanimity terms the results of Baker and Pixley can actually be used to derive a polynomial-time algorithm. We cover the (computationally) easier case of linear strong Mal'tsev conditions first in Section 3.1, before proving the more general result in Section 3.2.

3.1 A Polynomial-Time Result Using Near Unanimity Terms

In this section we show that the problem $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ is tractable if Σ is a linear strong Mal'tsev condition which implies the existence of a near unanimity term. We begin with a reminder of the definition of a near unanimity term.

Recall from Example 1.2.4: For $k \geq 3$ a *k-ary near unanimity term* is a term $m(x_1, \ldots, x_k)$ satisfying the k equations

$$m(x, x, \ldots, x, y, x, \ldots, x, x) \approx x.$$

for each position of the "lone dissenter" y. We use $\mathcal{NU}(k)$ to denote the strong Mal'tsev condition *"existence of a k-ary near unanimity term"* given by these equa-

tions. The Mal'tsev condition[1] $\mathcal{NU}$, *"existence of a near unanimity term"*, is defined to be the sequence $(\mathcal{NU}(k))_{k\geq 3}$. The study of near unanimity terms has a long history and was especially popularized with Baker and Pixley's celebrated result [2, Theorem 2.1]. We now reproduce the parts of that theorem upon which the main result of this section (as well as the main result of Section 3.2) will rely. Note that in [2] the result is stated at the level of varieties whereas for our purposes we have rephrased the theorem to consider particular algebras. The full theorem appears in Chapter 1 of this thesis as Theorem 1.2.3.

Theorem 3.1.1 (Corollary of [2], Theorem 2.1)**.** *Let* $\mathbf{A}$ *be an algebra,* $d \geq 2$ *and suppose that* $\mathbf{A}$ *has a* $(d+1)$*-ary near unanimity term. Then for any* $n \geq 1$ *and any partial operation* $f : D_f \to A$ *(*$D_f \subseteq A^n$*), if every subalgebra of* $\mathbf{A}^d$ *is closed under* f*, then* f *is the restriction of a term operation of* $\mathbf{A}$*.*[2]

Proof. This is an immediate consequence of the Baker-Pixley Theorem. See [2, Theorem 2.1]. □

As observed in [2, Section 2], this theorem of Baker and Pixley greatly simplifies the problem of determining whether or not a given (partial) operation is (the restriction of) a term operation of $\mathbf{A}$ in the case that $\mathbf{A}$ has a near unanimity term. The following result forms part of Lemma 3.1 in [2] and will give us a more precise algorithm for answering the same question. A proof of this result is an easy exercise in universal algebra.

Lemma 3.1.2 ([2], Lemma 3.1)**.** *Let* $\mathbf{A}$ *be an algebra,* $r, k \geq 1$*, and* $f : D_f \to A$ *a partial operation on* A *of arity* r*. The following are equivalent:*

1. *Every subalgebra of* $\mathbf{A}^k$ *is closed under* f*,*
2. *For any* $(r \times k)$*-matrix* M *over* A *with columns in* D_f*, the row vector* $f(M)$ *(whose entries are* f *applied to each column of* M*) is in the subalgebra of* $\mathbf{A}^k$ *generated by the rows of* M*.*

With this lemma and Theorem 3.1.1 in hand, we see that if $\mathbf{A}$ has a near unanimity term of arity $(d+1)$ then the problem of determining whether a given (partial) operation f on A of arity r is (the restriction of) a term operation of $\mathbf{A}$ is equivalent to the problem of determining whether for every $(r \times d)$-matrix M with columns in the domain of f, $f(M)$ is in the appropriate subalgebra of $\mathbf{A}^d$. We will use this result to show that instances of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ can be solved in polynomial-time if Σ is linear, strong, and implies the existence of a near unanimity term.

[1] See Definition 1.2.1.

[2] Following [2, Section 2], we say that $\mathbf{S} \leq \mathbf{A}^k$ is *closed under* f if for any choice of n k-tuples in S the k-tuple obtained by applying f coordinate-wise to these tuples is also in S whenever this tuple is defined.

Recall that our definition of "Σ *implies* $\mathcal{NU}$" is that for any algebra $\mathbf{A}$ we have: if $\mathbf{A}$ satisfies Σ, then $\mathbf{A}$ satisfies $\mathcal{NU}$. We may worry that the arity of the near unanimity term guaranteed by this implication could grow with the instance $\mathbf{A}$ of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$. Our next result rules out this potential problem.

Lemma 3.1.3. *Let Σ be a strong Mal'tsev condition which implies the existence of a near unanimity term. Then there is some $k \geq 3$ such that Σ implies the existence of a k-ary near unanimity term.*

Proof. Let Σ be a strong Mal'tsev condition which implies the existence of a near unanimity term and let $\mathcal{V} := \mathcal{V}(\Sigma)$ be the variety axiomatized by the equations of Σ. Let $\mathbf{F} := \mathbf{F}_{\mathcal{V}}(\{x, y\})$ be the free algebra in $\mathcal{V}$ on two generators.

Since $\mathbf{F} \in \mathcal{V}$ we have that $\mathbf{F}$ satisfies Σ. Hence $\mathbf{F}$ satisfies $\mathcal{NU}$.

I.e. there is some $k \geq 3$ such that $\mathbf{F}$ has a near unanimity term $m(x_1, \ldots, x_k)$ of arity k. For this term m we have:

$$m^{\mathbf{F}}(y, x, x, \ldots, x, x) = m^{\mathbf{F}}(x, y, x, \ldots, x, x) = \cdots = m^{\mathbf{F}}(x, x, x, \ldots, x, y) = x$$

and hence the equations

$$m(y, x, x, \ldots, x, x) \approx m(x, y, x, \ldots, x, x) \approx \cdots \approx m(x, x, x, \ldots, x, y) \approx x$$

hold in the variety $\mathcal{V}$.

Since for every algebra $\mathbf{A}$ we have that $\mathbf{A}$ satisfies Σ if and only if there are terms $t_1, \ldots, t_n$ of $\mathbf{A}$ such that the algebra $\mathbf{A}' := (A, t_1^{\mathbf{A}}, \ldots, t_n^{\mathbf{A}})$ lies in $\mathcal{V}$, and $\mathbf{A}' \in \mathcal{V}$ implies that $\mathbf{A}'$ satisfies $\mathcal{NU}(k)$, it follows that Σ implies $\mathcal{NU}(k)$, as required. □

Thanks to [37, Theorem 2.5] the decision problem *"does the strong Mal'tsev condition Σ imply $\mathcal{NU}$?"* is actually undecidable.[3] Nevertheless if there is some way to guarantee that the Mal'tsev condition in which we are interested actually implies $\mathcal{NU}$ (for example, if the equations of $\mathcal{NU}(k)$ are included in Σ), then we may use the results of Theorems 3.1.8 and 3.2.1 to solve the problem $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ or verify YES instances respectively. In practice for the Mal'tsev conditions of interest to universal algebraists it is already known whether or not they imply the existence of near unanimity terms.

We are now almost ready to prove the main result of this section. For convenience, we first introduce a lemma about linear strong Mal'tsev conditions in general. The result of the lemma is an observation which has been made many times before, for example in the proof of [22, Lemma 9.4]. The lemma is not a necessary ingredient in the proof of Theorem 3.1.8 but serves to shorten the proof and simplify complexity estimates (see the discussion after the proof of the theorem for details on bypassing Lemma 3.1.4).

[3] Take H in the statement of [37, Theorem 2.5] to be the set of strong Mal'tsev conditions Σ such that Σ implies $\mathcal{NU}$. The same argument shows that the decision problem *"does the strong Mal'tsev condition Σ imply $\mathcal{NU}(k)$?"* is also undecidable.

Lemma 3.1.4. *Let Σ be a linear strong Mal'tsev condition. Then there is a Mal'tsev condition T of the form $\{f(\overline{x}_1) \approx f(\overline{y}_1), \ldots, f(\overline{x}_l) \approx f(\overline{y}_l)\}$ (involving a single function symbol f) such that for any idempotent algebra $\mathbf{A}$ we have that $\mathbf{A}$ satisfies Σ if and only if $\mathbf{A}$ satisfies T.*

Proof. The idea is to compose all of the function symbols of Σ together in a particular way such that the original operations can be recovered using the idempotent law. We also replace any instance of the variable x (say) with the term $f(x, x, \ldots, x)$. See the proof of [22, Lemma 9.4] for more details. □

Example 3.1.5. Let Σ be the Mal'tsev condition

$$\Sigma := \{g(x,y) \approx g(y,x),\ g(x,y) \approx f(x,y,x),\ f(y,y,x) \approx f(x,x,y)\}.$$

An idempotent algebra $\mathbf{A}$ satisfies Σ if and only if it also satisfies the condition:

$$\Sigma_1 := \{h(xxx\ yyy) \approx h(yyy\ xxx),\ h(xxx\ yyy) \approx h(xyx\ xyx),\\ h(yyx\ yyx) \approx h(xxy\ xxy)\}$$

in which commas have been omitted and spaces added for readability. Term operations interpreting f and g can be recovered from a term operation interpreting h via $g^{\mathbf{A}}(x,y) := h^{\mathbf{A}}(xxx\ yyy)$ and $f^{\mathbf{A}}(x,y,z) := h^{\mathbf{A}}(xyz\ xyz)$. A term operation interpreting h can be recovered from idempotent term operations interpreting f and g via $h^{\mathbf{A}}(x_1,x_2,x_3,x_4,x_5,x_6) := f^{\mathbf{A}}(g^{\mathbf{A}}(x_1,x_4), g^{\mathbf{A}}(x_2,x_5), g^{\mathbf{A}}(x_3,x_6))$.

We now introduce a definition which will be used during the proof of Theorem 3.1.8.

Definition 3.1.6. Let A and X be sets and $r \geq 2$ an integer. For tuples $\overline{a} \in A^r$ and $\overline{x} \in X^r$ we say that $\overline{a}$ *matches the equality pattern of* $\overline{x}$ if for each $1 \leq i < j \leq r$ we have $\overline{x}|_i = \overline{x}|_j \implies \overline{a}|_i = \overline{a}|_j$.

Example 3.1.7. Let $X = \{w, x, y, z\}$ be a set of variables.

- The tuple $(0,0,1,1,1,0) \in \{0,1\}^6$ matches the equality pattern of
$$(w,w,x,x,y,z) \in X^6.$$
- Every tuple in $\{0,1\}^3$ matches the equality pattern of some $\overline{x}$ in
$$\{(y,x,x), (x,y,x), (x,x,y)\}.$$

Theorem 3.1.8. *Let Σ be a linear strong Mal'tsev condition which implies the existence of a near unanimity term. Then there is some $k \geq 3$ such that any instance $\mathbf{A}$ of Sat^{Id}_{Σ} can be solved by*

1. *Solving* $\mathbf{A}$ *as an instance of* $Sat^{Id}_{\mathcal{NU}(k)}$ *and then*

2. *Solving a corresponding instance* $P_{\mathbf{A}}$ *of CSP(*$\mathbf{A}$*) if* $\mathbf{A}$ *is a YES instance of* $Sat^{Id}_{\mathcal{NU}(k)}$.

Furthermore, the instance $P_{\mathbf{A}}$ *can be constructed by an algorithm whose runtime is polynomial in* $||\mathbf{A}||$*, and hence* $Sat^{Id}_{\Sigma} \in \mathsf{P}$.

Proof. Let Σ be a linear strong Mal'tsev condition which implies the existence of a near unanimity term and let $\mathbf{A}$ be an instance of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$. Without loss of generality[4] we may assume that Σ is a condition of the form

$$\{f(x^1_1, \dots, x^1_r) \approx f(y^1_1, \dots, y^1_r), \dots, f(x^l_1, \dots, x^l_r) \approx f(y^l_1, \dots, y^l_r)\}.$$

By Lemma 3.1.3 we know that Σ implies $\mathcal{NU}(k)$ for some fixed $k \geq 3$. Our first step then is to use the algorithm given by [24, Corollary 2.7 (1)] to determine whether $\mathbf{A}$ supports a k-ary near unanimity term. If $\mathbf{A}$ does not have a k-ary near unanimity term, then we return the answer NO, since $\mathbf{A}$ cannot satisfy Σ in this case.

Otherwise, we know that $\mathbf{A}$ has a k-ary near unanimity term. We will construct an instance $P_{\mathbf{A}} = (A^r, A, \mathcal{C}_{\mathbf{A}})$ of $\mathrm{CSP}(\mathbf{A})$ which has a solution if and only if $\mathbf{A}$ satisfies Σ. The instance $P_{\mathbf{A}}$ has variable set A^r (where r is the arity of the function symbol f) and domain A and hence a solution s to $P_{\mathbf{A}}$ is an r-ary operation on A. The constraints $\mathcal{C}_{\mathbf{A}}$ will be chosen such that:

- Any solution s to $P_{\mathbf{A}}$ is an operation which satisfies the equations of Σ,
- Any solution s to $P_{\mathbf{A}}$ is a term operation of $\mathbf{A}$, and
- If s is a term operation of $\mathbf{A}$ which satisfies the equations of Σ, then s is a solution to $P_{\mathbf{A}}$.

If we can choose constraints which guarantee these outcomes, then clearly $\mathbf{A}$ is a YES instance of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ if and only if $P_{\mathbf{A}}$ is a YES instance of $\mathrm{CSP}(\mathbf{A})$. We begin by defining the constraints that will guarantee that s satisfies the equations $s(x^\alpha_1, \dots, x^\alpha_r) \approx s(y^\alpha_1, \dots, y^\alpha_r)$ for each $\alpha \in \{1, \dots, l\}$.

Define

$$E := \{(a_1, \dots, a_r, b_1, \dots, b_r) \in A^{2r} \mid (\overline{a}, \overline{b}) \text{ matches the equality pattern of } (x^\alpha_1, \dots x^\alpha_r, y^\alpha_1, \dots y^\alpha_r) \text{ for some } \alpha = 1, \dots, l\}.$$

For each tuple $(\overline{a}, \overline{b}) \in E$ we include in $\mathcal{C}_{\mathbf{A}}$ the constraint $C_{\overline{a},\overline{b}} := ((\overline{a}, \overline{b}), =_A)$ which says that any solution s to $P_{\mathbf{A}}$ must satisfy $s(\overline{a}) = s(\overline{b})$. By guaranteeing the satisfaction of each of these equalities by any solution s to $P_{\mathbf{A}}$, we see that the equation $s(x^\alpha_1, \dots, x^\alpha_r) \approx s(y^\alpha_1, \dots, y^\alpha_r)$ is also satisfied by any solution.

[4] See Lemma 3.1.4

We now introduce the constraints which will guarantee that a solution s to $P_{\mathbf{A}}$ will be a term operation of $\mathbf{A}$. Using the theorem of Baker and Pixley (Theorem 1.2.3) we need only guarantee that every subalgebra of $\mathbf{A}^{k-1}$ is closed under any solution s, and using Lemma 3.1.2 we can encode this condition in the constraints of the instance $P_{\mathbf{A}}$, in the following manner:

For any choice $\overline{c}_1, \dots, \overline{c}_{k-1}$ of $k-1$ distinct tuples in A^r, we include in $\mathcal{C}_{\mathbf{A}}$ the constraint

$$T_{\overline{c}_1,\dots,\overline{c}_{k-1}} := \left((\overline{c}_1, \dots, \overline{c}_{k-1}), \left\langle \overline{d}_1, \dots, \overline{d}_r \right\rangle_{\mathbf{A}^{k-1}}\right)$$

where $\overline{d}_i$ is the i-th row of the matrix whose columns are (transposes of) $\overline{c}_1, \dots, \overline{c}_{k-1}$. To satisfy the constraint $T_{\overline{c}_1,\dots,\overline{c}_{k-1}}$ we must have $(s(\overline{c}_1), \dots, s(\overline{c}_{k-1})) \in \left\langle \overline{d}_1, \dots, \overline{d}_r \right\rangle_{\mathbf{A}^{k-1}}$. Since we include such a constraint for every choice of $\overline{c}_1, \dots, \overline{c}_{k-1}$ in A^r (the domain of the instance $P_{\mathbf{A}}$), we see that satisfying each of these constraints is equivalent to the second condition of Lemma 3.1.2. Using that lemma and Theorem 1.2.3, any solution to the constructed instance $P_{\mathbf{A}}$ is a term operation of $\mathbf{A}$.

On the other hand, it is clear that a term operation of $\mathbf{A}$ which satisfies the equations of Σ also satisfies all of the constraints outlined in the previous two paragraphs.

It follows that the instance $P_{\mathbf{A}} := (A^r, A, \mathcal{C}_{\mathbf{A}})$ of CSP($\mathbf{A}$) constructed above has a solution if and only if $\mathbf{A}$ is a YES instance of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$. Since $\mathbf{A}$ has a near unanimity operation it follows that $\mathbf{A}$ has "bounded width" ([28, Corollary 3.6], [34]) and hence CSP($\mathbf{A}$) is tractable via an algorithm whose runtime is independent of the algebra $\mathbf{A}$.[5] It remains only to prove that given $\mathbf{A}$, the instance $P_{\mathbf{A}}$ can be built in polynomial-time.

The equality constraint relation $=_A$ can clearly be constructed in time $O(|A|)$ and need only be constructed once. To build all of the constraints $C_{\overline{a},\overline{b}}$ for $\overline{a}, \overline{b} \in E$ we need to check for each tuple $(a_1, \dots, a_r, b_1, \dots, b_r) \in A^{2r}$ whether we have $(a_1, \dots, a_r, b_1, \dots, b_r) \in E$ or not. For a fixed tuple $(a_1, \dots, a_r, b_1, \dots, b_r)$ this procedure is clearly linear in r and independent of $||\mathbf{A}||$. Hence the time taken to build all of the constraints $C_{\overline{a},\overline{b}} \in E$ is $O(|A|^{2r})$ which is $O(||\mathbf{A}||^{2r})$.

Next, for each choice of distinct $\overline{c}_1, \dots, \overline{c}_{k-1} \in A^r$ we need to build the corresponding constraint relation $\left\langle \overline{d}_1, \dots, \overline{d}_r \right\rangle_{\mathbf{A}^{k-1}}$ for the constraint $T_{\overline{c}_1,\dots,\overline{c}_{k-1}}$ defined above. Thanks to [20, Proposition 6.1] this relation can be built in time $O(||\mathbf{A}||^k)$. Since there are ${}^{|A|^r}P_{k-1} = \frac{|A|^r!}{(|A|^r-(k-1))!}$-many such constraints to construct, the total time to build all of the constraints $T_{\overline{c}_1,\dots,\overline{c}_{k-1}}$ is $O((|A|^r)^{(k-1)}||\mathbf{A}||^k)$ which is $O(||\mathbf{A}||^{r(k-1)+k})$.

Since $k \geq 3$ we see that the total time taken to construct the instance $P_{\mathbf{A}}$ is $O(||\mathbf{A}||^{r(k-1)+k})$, a polynomial in $||\mathbf{A}||$, as required. □

Example 3.1.9. As an example of a Mal'tsev condition which satisfies the hypotheses of Theroem 3.1.8, consider the Mal'tsev condition of *having a symmetric majority term*

[5] See Sections 5.3, 5.5, and 5.6 of the survey article [5] for an overview on algebras of bounded width.

given by the equations:

$$\{m(x,y,z) \approx m(y,x,z),$$
$$m(x,y,z) \approx m(y,z,x),$$
$$m(x,x,y) \approx m(x,y,x) \approx m(y,x,x) \approx x\}.$$

This condition is satisfied in any lattice by the term

$$m(x,y,z) := (x \wedge y) \vee (y \wedge z) \vee (z \wedge x)$$

and is strictly weaker than the condition of having lattice terms since that condition is known to be nonlinear (see [41] for a proof). Theorem 3.1.8 tells us that the existence of a symmetric majority term can be checked in polynomial-time for finite idempotent algebras whereas currently the best algorithms for detecting lattice terms are in the class NP (Corollary 3.2.3).

The runtime of the algorithm presented in the proof of Theorem 3.1.8 is a little obscure since it depends on the method used to solve the constructed instance of CSP($\mathbf{A}$). In [10, Section 3.1] an algorithm is given to transform any given instance P of CSP($\mathbf{A}$) to a so-called "l-minimal" instance P' (where $l \geq 2$ is some fixed integer). "l-minimal" here is a consistency requirement for the instance P'. The result of [5, Theorem 68] implies that it is enough to implement the algorithm given in [10, Section 3.1] to transform the instance $P_{\mathbf{A}}$ built in the proof of Theorem 3.1.8 into a corresponding 3-minimal instance $P'_{\mathbf{A}}$, rejecting only if the instance $P'_{\mathbf{A}}$ has any empty constraint relations.[6]

Using the algorithm given in [10, Section 3.1] the instance $P'_{\mathbf{A}}$ can be built in time

$$O(m^3(|A|^r)^3)$$

where m is the total number of tuples appearing in constraint relations of $P_{\mathbf{A}}$. Thus m is

$$O(|A|^{2r}(|A|^2) + (|A|^r)^{k-1}|A|^{k-1}) = O(|A|^{(r+1)(k-1)})$$

for a total time requirement for establishing 3-minimality in the order of

$$O(|A|^{3(r+1)(k-1)+3r}).$$

Note that the proof of Theorem 3.1.8 uses the idempotence of the algebra $\mathbf{A}$ in two ways. The first use of idempotence is to write the Mal'tsev condition Σ as a condition involving only one function symbol. This is purely to simplify the presentation of the

[6] The notions of "width $(2,3)$" from [5] and "3-minimal" from [10] are formally different conditions but both can be seen to imply a level of consistency sufficient to guarantee that partial solutions can be extended to solutions.

proof of the theorem. Indeed, in practice it would be more efficient[7] to construct a slightly different instance of CSP($\mathbf{A}$) which has domain $A^{r_1} \sqcup A^{r_2} \sqcup \cdots \sqcup A^{r_n}$ where $r_1, \ldots, r_n$ are the arities of the function symbols involved in the equations of the condition Σ. In this case, a solution to the given instance could be regarded as a family of operations on A and equality constraints may be placed between the different "summands" of the disjoint union to guarantee satisfaction of the Mal'tsev condition Σ (if necessary).[8] Constraints can also be used to guarantee that each operation individually is a term operation of $\mathbf{A}$ using the theorem of Baker and Pixley in an analogous manner to the proof presented.

The second use of idempotence is in checking whether $\mathbf{A}$ satisfies the condition $\mathcal{NU}(k)$. The complexity of the decision problem $\mathrm{Sat}_{\mathcal{NU}(k)}$ remains an open question in the field. We remark now that if the problem $\mathrm{Sat}_{\mathcal{NU}(k)}$ is demonstrated to be tractable (for arbitrary fixed k), then the result of the theorem presented above can be extended to the problem Sat_Σ for any linear strong Mal'tsev condition Σ which implies $\mathcal{NU}(k)$, removing any need for the assumption of idempotence. We summarize this in a corollary:

Corollary 3.1.10. *Let Σ be a linear strong Mal'tsev condition which implies the existence of a near unanimity term. Then there is some $k \geq 3$ such that any instance $\mathbf{A}$ of Sat_Σ can be solved by*

1. *Solving $\mathbf{A}$ as an instance of $Sat_{\mathcal{NU}(k)}$ and then*
2. *Solving a corresponding instance $P_\mathbf{A}$ of CSP($\mathbf{A}$) if $\mathbf{A}$ is a YES instance of $Sat_{\mathcal{NU}(k)}$.*

Furthermore, the instance $P_\mathbf{A}$ can be constructed by an algorithm whose runtime is polynomial in $||\mathbf{A}||$, and hence if $Sat_{\mathcal{NU}(k)}$ is in P then so is Sat_Σ.

We also highlight the following corollary which deals with the search problem related to a given Σ-satisfaction problem. Further results of this kind (for various Σ) are found in Chapter 4.

Corollary 3.1.11. *Let Σ be a linear strong Mal'tsev condition which implies $\mathcal{NU}$. There is a polynomial-time algorithm which takes as input a finite idempotent algebra $\mathbf{A}$ and returns the operation tables of term operations of $\mathbf{A}$ witnessing that $\mathbf{A}$ satisfies Σ whenever such terms exist (and otherwise returns the answer NO).*

[7] By invoking Lemma 3.1.4 we construct an instance $P_\mathbf{A}$ whose domain may have size as large as $|A|^R$ where $R = \prod_{i=1}^{n} r_i$, whereas the disjoint union constructed above has size $\sum_{i=1}^{n} |A|^{r_i}$.

[8] We may also need constraints of the form $((a_1, \ldots, a_r), \{a\})$ to guarantee the satisfaction of equations of height < 1 (see Definition 5.1.1). If the input algebra is not idempotent then the subset $\{a\}$ may not be a subuniverse of $\mathbf{A}$ which is a technical violation of our definition of CSP($\mathbf{A}$). The result of the theorem still holds in this context because the subset $\{a\}$ is a subuniverse of the (idempotent) algebra $\mathbf{B} := (A, m^\mathbf{A}(x_1, \ldots, x_k))$ whose basic operation is the k-ary near unanimity term operation of $\mathbf{A}$, and the instance of CSP($\mathbf{A}$) constructed can be regarded instead as an instance of CSP($\mathbf{B}$).

Proof. Let $P_{\mathbf{A}}$ be the instance of CSP($\mathbf{A}$) constructed in the proof of Theorem 3.1.8. If there is no solution to this instance then the algorithm halts and outputs the answer NO. Otherwise, a folklore technique in constraint satisfaction[9] allows us to construct a solution by incrementally constraining each variable to take one particular value and determining whether a solution exists in which this variable takes on that particular value. Taking each variable in turn we search the possible values which that variable could take on and when we find a value for which there is a solution we keep this assignment as a new constraint and move on to the next variable. There are $|A|^r$-many variables in the instance and $|A|$-many possible values for each variable. Hence we need to solve at most $|A|^r|A| - 1 = O(|A|^{r+1})$-many instances of CSP($\mathbf{A}$) in order to determine a solution to $P_{\mathbf{A}}$. As observed in the proof of Theorem 3.1.8 this solution is a term operation of $\mathbf{A}$ witnessing that $\mathbf{A}$ satisfies Σ. □

In the next section we will consider nonlinear strong Mal'tsev conditions. The proof of Theorem 3.1.8 relied heavily on reducing an instance $\mathbf{A}$ of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ to an instance $P_{\mathbf{A}}$ of CSP($\mathbf{A}$). For nonlinear strong Mal'tsev conditions it remains unclear whether a similar construction can work to achieve a polynomial-time algorithm. However the theorem of Baker and Pixley still provides an efficient algorithm to determine whether a given operation is a term operation of $\mathbf{A}$ (in the case that $\mathbf{A}$ has a near unanimity term). This result is used in the next section to show that the satisfaction of nonlinear strong Mal'tsev conditions which imply the existence of a near unanimity term can at least be verified in polynomial-time, placing this problem in NP.

3.2 An NP Result Using Near Unanimity Terms

Until recently the only known results on the complexity of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ for nonlinear strong Mal'tsev conditions Σ were hardness results demonstrating that these problems are EXPTIME-complete (see for example [19], [20]). In this section we prove that when Σ implies the existence of a near unanimity term, then the problem $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ is in NP. In particular, this provides examples of nonlinear Mal'tsev conditions whose idempotent satisfaction problem is not EXPTIME-complete (assuming NP $\neq$ EXPTIME). Such an example is given in Corollary 3.2.3 after the statement and proof of the main result.

Theorem 3.2.1. *Let Σ be a strong Mal'tsev condition which implies the existence a near unanimity term. Then the problem Sat^{Id}_{Σ} is in NP.*

Proof. Let $\mathbf{A}$ be an instance of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$. Since Σ implies $\mathcal{NU}$, using Lemma 3.1.3 it follows that Σ implies $\mathcal{NU}(k)$ for some $k \geq 3$. As in the linear case, our first step is to use the algorithm of Horowitz [24, Corollary 2.7 (1)] to determine whether or not $\mathbf{A}$ has a k-ary near unanimity term. If the algorithm returns the answer NO, then it follows that $\mathbf{A} \nvDash \Sigma$ and we are done.

[9] This technique is alluded to for example at the very end of [5, Section 3.2].

Otherwise, we know that $\mathbf{A}$ supports a near unanimity term of arity k. Let $h_1, \ldots, h_m$ be a complete list of the operation symbols found in the equations of Σ. We show that given (tables for) operations $g_1, \ldots, g_m$ with arity(g_i) =arity(h_i) for each $i = 1, \ldots, m$, it can be verified in polynomial-time that $g_1, \ldots, g_m$ are term operations of $\mathbf{A}$ witnessing that $\mathbf{A}$ satisfies Σ.

Let $\sigma(\overline{x}) \approx \tau(\overline{y})$ be an equation of Σ, and let s, t denote the operations obtained from σ and τ respectively by interpreting each occurrence of h_i in σ and τ as the function g_i for each $i \leq m$. Let p denote the number of distinct variables occurring in the equation $\sigma(\overline{x}) \approx \tau(\overline{y})$ and note that p is independent of the algebra $\mathbf{A}$. The operation tables for s and t can both be obtained in time $O(|A|^p)$ since for each tuple in A^p we need to look up a constant number of values from some of the tables g_i (for $i \in I \subseteq [m]$ determined by the structure of the terms σ and τ). The equality of the functions s and t can be verified by checking that for each $\overline{a} \in A^p$ we have $s(\overline{a}) = t(\overline{a})$.[10] Having built and stored the tables for s and t this equality can be checked in constant time (for a fixed tuple $\overline{a}$).

Hence a single equation can be verified (given the tables for g_i) in time $O(||\mathbf{A}||^p)$ where p is the number of variables occurring in the equation. Since the number of equations in Σ and the arities of the associated term operations are both independent of the algebra $\mathbf{A}$, it follows that all the equations of Σ can be verified in polynomial-time.

It remains to see that for each $1 \leq i \leq m$ we can verify in polynomial-time that g_i is a term operation of $\mathbf{A}$. Using the results of Baker and Pixley (Theorem 3.1.1 and Lemma 3.1.2), it suffices to check that for every $\overline{c}_1, \ldots, \overline{c}_{k-1} \in A^n$ (where $n =$ arity(g_i)) we have $(g(\overline{c}_1), \ldots, g(\overline{c}_{k-1})) \in \left\langle \overline{d}_1, \ldots, \overline{d}_n \right\rangle_{\mathbf{A}^{k-1}}$, where $\overline{d}_j = (\overline{c}_1|_j, \ldots, \overline{c}_{k-1}|_j)$ is the transpose of the jth column of the matrix whose rows are $\overline{c}_1, \ldots, \overline{c}_{k-1}$. For each choice of distinct tuples $\overline{c}_1, \ldots, \overline{c}_{k-1}$ this condition can be checked in time $O(||\mathbf{A}||^k)$ by [20, Proposition 6.1]. Since there are ${}^{|A|^r}P_{k-1} = \frac{|A|^r!}{(|A|^r-(k-1))!}$ choices for the tuples $\overline{c}_1, \ldots, \overline{c}_{k-1}$, verifying that g_i is a term operation of $\mathbf{A}$ is achieved in time $O(|A|^{n(k-1)}||\mathbf{A}||^k)$ which is $O(||\mathbf{A}||^{n(k-1)+k})$.

Let r be the maximum arity of all the g_i (for $1 \leq i \leq m$). Then it follows that to verify that every g_i is a term operation of $\mathbf{A}$ takes time $O(||\mathbf{A}||^{r(k-1)+k})$ (since m is fixed independently of $\mathbf{A}$).

Given a YES instance $\mathbf{A}$ of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ we define a certificate $C(\mathbf{A})$ for $\mathbf{A}$ as a string of operation tables for term operations $g_1, \ldots, g_m$ of $\mathbf{A}$ witnessing that $\mathbf{A}$ satisfies Σ. A polynomial-time verifier[11] for $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ is then given by the following algorithm whose input is $(\mathbf{A}, C(\mathbf{A}))$

1. Solve $\mathbf{A}$ as an instance of $\mathrm{Sat}^{\mathrm{Id}}_{\mathcal{NU}(k)}$ for an appropriate choice of k (independent of $\mathbf{A}$) and obtain the answer YES, in polynomial-time thanks to [24, Corollary

[10] It may be that not every variable occurring in the equation $\sigma(\overline{x}) \approx \tau(\overline{y})$ occurs in both of the terms σ and τ. For example in the equation $\sigma(x, y) \approx \tau(x, y, z)$. In this case we have $p = 3$ and for $\overline{a} = (a, b, c) \in A^3$ we consider $s(\overline{a})$ to be $s(a, b)$. This is simply for notational convenience.

[11] See Definition 2.1.4.

2.7(1)].

2. Verify that the equations of Σ hold when each h_i is interpreted by the function g_i, in the manner described above, in time $O(||\mathbf{A}||^p)$.

3. Verify that each g_i is a term operation of $\mathbf{A}$, in the manner described above, in time $O(||\mathbf{A}||^{r(k-1)+k})$.

Since each of these items can be achieved in polynomial-time (as outlined above), it follows that $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ is in NP. □

The only use of idempotence in the above proof is to determine whether or not $\mathbf{A}$ is a YES instance of $\mathrm{Sat}^{\mathrm{Id}}_{\mathcal{NU}(k)}$. It follows that if the problem $\mathrm{Sat}_{\mathcal{NU}(k)}$ can be shown to be in NP (for arbitrary fixed k), then this would imply the corresponding result $\mathrm{Sat}_{\Sigma} \in \mathsf{NP}$ for any Σ which implies $\mathcal{NU}$. As in the previous section, we summarize this observation in the following corollary:

Corollary 3.2.2. *Fix $k \geq 3$. If $Sat_{\mathcal{NU}(k)}$ is in NP then so is Sat_{Σ} for any strong Mal'tsev condition Σ which implies $\mathcal{NU}(k)$.*

In particular, if idempotence can be removed from the hypothesis of Theorem 5.2.4, then we can also remove idempotence from Theorem 3.2.1.

The corollaries that follow are novel consequences of Theorem 3.2.1.

Corollary 3.2.3. *Let Λ be the Mal'tsev condition of having lattice terms. Then $Sat^{Id}_{\Lambda} \in$ NP.*

Proof. If $\vee$ and $\wedge$ are terms of an algebra $\mathbf{A}$ which satisfy the lattice identities, then the term $m(x, y, z) := (x \vee y) \wedge (y \vee z) \wedge (x \vee z)$ is clearly a majority term of $\mathbf{A}$. It follows that Λ implies $\mathcal{NU}$ and hence $\mathrm{Sat}^{\mathrm{Id}}_{\Lambda} \in \mathsf{NP}$ by Theorem 3.2.1. □

Corollary 3.2.4. *If NP $\neq$ EXPTIME, then there is a nonlinear Mal'tsev condition whose idempotent satisfaction problem is not EXPTIME-complete.*

Proof. It is known that the condition of having lattice terms is not equivalent to a linear Mal'tsev condition (see for example [41]). Hence there is a nonlinear Mal'tsev condition whose idempotent satisfaction problem is in NP, as required. □

It is interesting to contrast the result of Corollary 3.2.3 with that of [19, Theorem 4.5] which says that detection of a semilattice term is EXPTIME-complete even for idempotent algebras. So far there are very few results concerning the complexity of MCSPs for nonlinear Mal'tsev conditions. It seems intuitive that having "more structure" (e.g. implying $\mathcal{NU}$) is easier to detect than conditions without such strong structural consequences. The following conjecture is grounded within that ethos but represents a much more attainable goal. The Mal'tsev condition described is a known weakening of the semilattice condition.

Conjecture 3.2.5. *The problem $Sat^{Id}_{2\text{-}semi}$ is EXPTIME-complete, where 2-semi is the (nonlinear*[12]*) Mal'tsev condition of having a 2-semilattice term:*

$$2\text{-}semi := \{b(x,x) \approx x,\ b(x,y) \approx b(y,x),\ b(x,b(x,y)) \approx b(x,y)\}$$

The following two conjectures are also in keeping with the ethos described above and the first is an obvious consequence of the second.

Conjecture 3.2.6. *The problem Sat^{Id}_{Λ} is in P where Λ is the condition of having lattice terms.*

Conjecture 3.2.7. *The problem Sat^{Id}_{Σ} is in P whenever Σ is a strong Mal'tsev condition which implies $\mathcal{NU}$.*

[12]It can be demonstrated that the condition 2-semi is not preserved under retractions and hence it follows from the result of [6, Proposition 5.3] that this condition is indeed nonlinear.

Chapter 4

On the Construction of Term Operations

Until this point we have primarily been concerned with the complexity of the decision problem: does $\mathbf{A}$ satisfy the Mal'tsev condition $\mathcal{M}$? I.e. do there exist terms in the language of $\mathbf{A}$ such that the associated term operations satisfy the equations of $\mathcal{M}$? We now turn our attention to the complexity of a related search problem: given a finite algebra $\mathbf{A}$, construct the operation tables of term operations of $\mathbf{A}$ which satisfy the Mal'tsev condition $\mathcal{M}$ if such terms exist.

In this chapter, we introduce a recent result of Kazda, Opršal, Valeriote and Zhuk [29] which states that the problem of building Mal'tsev term operations is tractable for finite idempotent algebras. We then extend this result to include edge term operations of any fixed arity (as well as several other conditions). This result for edge term operations will be used later in order to derive the NP result of Chapter 5. We also explore other well-known conditions for which the problem of obtaining operation tables is tractable and suggest a conjecture that for any condition whose satisfaction can be decided in polynomial-time there is a corresponding polynomial-time algorithm to construct the relevant term operations. We begin with the relatively simple problem of building operation tables for cyclic term operations as a guide to understanding the basic proof technique. We then reproduce the proof from [29] to serve as a guide for the new result Theorem 4.4.5. In the final section of this chapter, we look at more complicated conditions involving multiple operation symbols and formulate conjectures for future investigation.

4.1 Building Local Term Operations

As we will see throughout this chapter, obtaining operation tables or circuits for term operations satisfying certain equations will typically involve subalgebra generation in order to obtain some term operations satisfying a "local" version of the given equations. The following proposition is an extension of [20, Proposition 6.1] and was

first observed in [29, Section 4]. This result will be used in almost all subsequent results of this chapter, and hence we find it convenient to formalize it here.

Proposition 4.1.1. *Fix $k, n \geq 1$. There is a polynomial-time algorithm*[1] *which takes as input a finite algebra $\mathbf{A}$, together with tuples $\overline{a}_1, \ldots, \overline{a}_n \in A^k$, and returns the subuniverse B of $\mathbf{A}^k$ generated by $\{\overline{a}_1, \ldots, \overline{a}_n\}$ together with, for each $\overline{b} \in B$, the operation table $t_{\overline{b}}(x_1, \ldots, x_n)$ of a term operation of $\mathbf{A}$ such that $t_{\overline{b}}(\overline{a}_1, \ldots, \overline{a}_n) = \overline{b}$.*

Proof. Let k and n be fixed. We define an algorithm which takes input $\mathbf{A}, \overline{a}_1, \ldots, \overline{a}_n$ and generates every element $\overline{b} \in \mathbf{B} := \langle \overline{a}_1, \ldots, \overline{a}_n \rangle_{\mathbf{A}^k}$ while also storing, for each $\overline{b}$ generated, the operation table of a term operation $t_{\overline{b}}$ of $\mathbf{A}$ which generates the element $\overline{b}$ when applied to the generators $\overline{a}_1, \ldots, \overline{a}_n$. The algorithm is defined as follows:

[1] We consider n and k to be fixed here- the algorithm described runs in polynomial-time with respect to $||\mathbf{A}||$.

Algorithm 4.1: Generating a Subpower and Storing Term Operations

Input: A finite algebra $\mathbf{A} := (A, \mathcal{F})$ and $\overline{a}_1, \ldots, \overline{a}_n$, tuples in A^k
Output: The subuniverse of A^k generated by $\overline{a}_1, \ldots, \overline{a}_n$ together with the operation table of a term operation $t_{\overline{b}}$ for every element $\overline{b}$ in the generated subpower, which satisfies the equality $t_{\overline{b}}(\overline{a}_1, \ldots, \overline{a}_n) = \overline{b}$.

1: $B_0 := \{\overline{a}_1, \ldots, \overline{a}_n\}$
2: **for** each $i = 1, \ldots, n$ **do**
3: **store** $t_{\overline{a}_i}(x_1, \ldots, x_n) := x_i$
4: **endfor**
5: $j := 0$
6: **while** $B_j \neq \emptyset$ **do**
7: $B_{j+1} := \emptyset$
8: **for** each $f \in \mathcal{F}$ **do**
9: $r_f := \text{arity}(f)$
10: **for** each $\overline{c} = (\overline{c}_1, \ldots, \overline{c}_{r_f}) \in (\bigcup_{i=0}^{j} B_i)^{r_f} \setminus (\bigcup_{i=0}^{j-1} B_i)^{r_f}$ **do**
11: **if** $f(\overline{c}) \in \bigcup_{i=0}^{j} B_i$ **then**
12: **go to next** $\overline{c}$
13: **else**
14: **add** $f(\overline{c})$ to B_{j+1}
15: **store** $t_{f(\overline{c})}(x_1, \ldots, x_n) := f(t_{\overline{c}_1}(x_1, \ldots, x_n), \ldots, t_{\overline{c}_{r_f}}(x_1, \ldots, x_n))$
16: **endif**
17: **endfor**
18: **endfor**
19: $j := j + 1$
20: **endwhile**
21: $B := \bigcup_{i=0}^{j} B_j$
22: **return** $\{(\overline{b}, t_{\overline{b}}) \mid \overline{b} \in B\}$.

Correctness of the algorithm follows from standard results in universal algebra (see for example [15, Chapter II, Theorem 3.2]). In fact, as suggested in the proof of [20, Proposition 6.1], what this algorithm does is to apply every basic operation f of $\mathbf{A}$ to every tuple in B^{r_f} without ever applying a basic operation to the same tuple twice. Lines (11)-(15) of the algorithm are therefore executed at most $|\mathcal{F}| \cdot ||\mathbf{A}||^k$-many[2] times and this clearly dominates the computation.

Calculating $f(\overline{c})$ takes time $O(kr_f)$ since $\overline{c}$ is a k-tuple of inputs to f and we need to read each one and then use a constant-time lookup in the table for f. We can

[2]Recall from Definition 1.1.1 that $||\mathbf{A}|| = \sum_{f \in \mathcal{F}} |A|^{\text{arity}(f)}$.

then check the condition of the **if** statement in line (11) in time $O(1)$. Assuming the condition fails, we require $O(r_f|A|^n)$-time to construct the table for $t_{f(\overline{c})}(x_1, \ldots, x_n)$ in line (15), since for each tuple $\overline{d} \in A^n$ we need to look up the values of $t_{\overline{c}_1}(\overline{d}), \ldots, t_{\overline{c}_{r_f}}(\overline{d})$ and then read this tuple and look up the value of f at this tuple. Each of these lookups is achieved in constant time since we have already stored the relevant operation tables.

Let R denote the maximum arity of all the operations in $\mathcal{F}$. Then the algorithm correctly produces all elements of the subuniverse B and the corresponding operation tables in time

$$O(|\mathcal{F}| \cdot ||\mathbf{A}||^k \cdot (kR + 1 + R|A|^n))$$

which is $O(||\mathbf{A}||^{k+n+2})$, a polynomial in $||\mathbf{A}||$ as required. □

It follows from the lemma that given a finite algebra $\mathbf{A}$ together with $\overline{a}_1, \ldots, \overline{a}_n$ and $\overline{a} \in A^k$ we can quickly (i.e. in polynomial-time with respect to $||\mathbf{A}||$) build the operation table of a term operation $t_{\overline{a}}$ satisfying $t_{\overline{a}}(\overline{a}_1, \ldots, \overline{a}_n) = \overline{a}$ if such a term operation exists (and otherwise answer NO). Simply halt the above algorithm when the tuple $\overline{a}$ is generated, or alternatively, complete the algorithm and search the output for $\overline{a}$. Similarly, if $(x_1, \ldots, x_k) \in X^k$ is a fixed tuple of variables from a set X, then we can find an element $\overline{a} \in \langle \overline{a}_1, \ldots, \overline{a}_n \rangle$ which matches the equality pattern[3] of $\overline{x}$ and build the operation table of a term operation $t_{\overline{a}}$ of $\mathbf{A}$ which satisfies

$$t_{\overline{a}}(\overline{a}_1, \ldots, \overline{a}_n) = \overline{a}$$

or else determine that no such $\overline{a}$ exists in polynomial-time with respect to $||\mathbf{A}||$.

Before moving on to the main results of the chapter, we briefly discuss the related problem of building circuits for term operations of $\mathbf{A}$. In this thesis we have elected to present direct proofs that the operation tables of certain term operations may be computed efficiently. An operation table for a given term operation can also be viewed as a 1-gate circuit[4] for computing the term, in the extended language where that term operation is included as one of the basic operations. On the other hand, if f is an operation on A of fixed arity r and a circuit with maximum gate arity R of size K is given for computing f, then a table for f can be computed in time $O(RK|A|^r)$ by simply running the circuit on every possible input and recording all of the input-output pairs as a table. From a practical perspective then, if we wish to obtain values of a term operation f of $\mathbf{A}$ (where the arity of f is fixed independently of $\mathbf{A}$), then a polynomial-size circuit for f is as convenient as an operation table for f (up to polynomial-time reductions).

The following proposition is an analogue of Proposition 4.1.1 which will allow us to derive circuits for term operations in the same way that Proposition 4.1.1 allows us to build operation tables for term operations. The result is essentially a porism of [29, Theorem 6]. Readers who prefer the circuit-first approach may view Proposition

[3] See definition 3.1.6.
[4] See Definition 2.4.1.

4.1.1 as a corollary of Proposition 4.1.2. From the perspective outlined in the previous paragraph, they may be viewed as morally equivalent results.

Proposition 4.1.2. *Fix $k, n \geq 1$. There is a polynomial-time algorithm which takes as input a finite algebra $\mathbf{A}$, together with tuples $\overline{a}_1, \ldots, \overline{a}_n \in A^k$, and returns the set*

$$\{(\overline{b}, m_{\overline{b}}) \mid \overline{b} \in \mathbf{B} := \left\langle \overline{a}_1, \ldots, \overline{a}_n \right\rangle_{\mathbf{A}^k} \text{ and}$$
$$\overline{b} \text{ is the } m_{\overline{b}}\text{-th element of } \mathbf{B} \text{ generated by the algorithm}\}$$

together with a circuit $C_{\mathbf{B}}$ in the language of $\mathbf{A}$ with n inputs and $|B|$-many designated outputs such that the value of the $m_{\overline{b}}$-th output of the circuit $C_{\mathbf{B}}$ on input $\overline{a}_1, \ldots, \overline{a}_n$ is $\overline{b}$.

Proof. Simply adapt Algorithm 4.1 in the following ways:

1. Rather than storing the operation table $t_{\overline{a}_i}(x_1, \ldots, x_n) = x_i$ on line (3), replace the **for** loop in lines (2)-(4) with the single instruction: construct and store the circuit C_n defined to be the circuit with inputs $x_1, \ldots, x_n$ and designated outputs $x_1, \ldots, x_n$ (and no gates).

2. On line (15), rather than storing the operation table $t_{f(\overline{c})}(x_1, \ldots, x_n)$, we construct and store the circuit $C_{m_{f(\overline{c})}}$. Suppose $f(\overline{c})$ is the m-th element generated so far.[5] We follow Step 1 in the proof of [29, Theorem 6] to construct the circuit C_m. C_m is a circuit with inputs $x_1, \ldots, x_n$ and with m outputs such that if the tuples $\overline{a}_1, \ldots, \overline{a}_n$ are inputted into C_m, then the m outputs are the first m elements of $\left\langle \overline{a}_1, \ldots, \overline{a}_n \right\rangle_{\mathbf{A}^k}$ generated by our algorithm in the same order in which they are generated. C_m is constructed from C_{m-1} inductively (for $n+1 \leq m \leq |B|$) via appending one gate, labeled f, whose inputs are those outputs of C_{m-1} corresponding to the previously generated tuples $\overline{c}_1, \ldots, \overline{c}_{r_f}$ and this gate is designated as the m-th output of C_m (the first $m-1$ outputs of C_m are the $m-1$ outputs of C_{m-1}). Clearly the size of C_m (for $m \geq n$) is m, and C_m has the correct output values described on the inputs $\overline{a}_1, \ldots, \overline{a}_n$. We note here that the time taken to build C_m from C_{m-1} is $O(r_f)$ since we simply add r_f new edges and one new vertex which we designate as the m-th output.

3. On line (22), return the set $\{(\overline{b}, m_{\overline{b}}) \mid \overline{b} \in B\}$ and the circuit $C_{\mathbf{B}} := C_{|B|}$.

The time requirement for the adapted algorithm (based on the analysis found in the proof of Proposition 4.1.1) is $O\left(|\mathcal{F}| \cdot ||\mathbf{A}||^k \cdot (kR + 1 + R)\right)$ where R is the maximum arity of the basic operations of $\mathbf{A}$. Hence the adapted algorithm correctly produces the desired output in time $O(||\mathbf{A}||^{k+2})$, a polynomial in $||\mathbf{A}||$, as required. □

[5] Keeping track of the order in which the elements are generated will clearly not affect the asymptotic complexity of our algorithm. We adopt the convention that $\overline{a}_1, \ldots, \overline{a}_n$ are the first n elements generated and hence we are now assuming $m \geq n+1$.

The time estimates obtained in Propositions 4.1.1 and 4.1.2 suggest the following improvement on the time taken to obtain the desired operation tables given by Algorithm 4.1:

1. Run the adapted algorithm described in the proof of Proposition 4.1.2 to obtain the circuit $C_{\mathbf{B}}$ which computes all of the relevant term operations.

2. Run the circuit $C_{\mathbf{B}}$ on all the tuples in A^n and record input-output pairs for each of the term operations calculated by the circuit.

Since the circuit $C_{\mathbf{B}}$ has size $|A|^k$, the time taken to complete these two steps is

$$O\left(|\mathcal{F}| \cdot ||\mathbf{A}||^k \cdot (kR + 1 + R)\right) + O\left(R|A|^n|A|^k\right)$$

which our crude estimates render as $O(||\mathbf{A}||^{k+n+1})$. While this is in theory an improvement on the estimate obtained by our analysis in Proposition 4.1.1 we remark here that the actual time taken for the calculation will depend heavily on the implementation of the algorithm in specific operational contexts. In order to obtain accurate worst-case analysis in subsequent results of this chapter, we use the time estimate $O(||\mathbf{A}||^{k+n+2})$ whenever we use the result of these propositions to obtain operation tables for term operations satisfying specific local conditions (e.g. in Theorems 4.2.1, 4.3.1, and 4.4.5).

4.2 Building a Cyclic Operation

Recall from Example 1.2.5: a *cyclic term of arity k* is a term $c(x_1, \ldots, x_k)$ satisfying the equation[6]

$$c(x_1, x_2, \ldots, x_k) \approx c(x_2, \ldots, x_k, x_1)$$

The result of [4, Lemma 2.4] clearly implies that the problem $\mathrm{Sat}^{\mathrm{Id}}_{\mathcal{C}(k)}$ is tractable where $\mathcal{C}(k)$ is the Mal'tsev condition of having a cyclic term of arity k. The next result shows that obtaining an operation table for such a term operation is also a tractable problem. The proof is based on an algorithm for the k-ary cyclic term satisfaction problem $\mathrm{Sat}^{\mathrm{Id}}_{\mathcal{C}(k)}$ which was presented to the author of this thesis by his supervisor, Matt Valeriote. Original authorship of that algorithm is attributed to Valeriote and Willard [43] and can also be viewed as a porism of [4, Lemma 2.4].

[6]We follow [5, Subsection 4.4] in our definition of cyclic terms. It may be more traditional (see for example [4]) to require that cyclic terms also satisfy the equation of being idempotent ($c(x, \ldots, x) \approx x$). In this case, the result of Theorem 4.2.1 is only valid for idempotent algebras (for which the constructed term operation is necessarily idempotent) and the corresponding decision problem is actually EXPTIME-complete in general thanks to [23, Corollary 5.1.9]. One significant difference between the two definitions is that using the definition presented here, any algebra with a constant term operation has cyclic operations of all arities (since a constant operation satisfies all height 1 identities).

Theorem 4.2.1. *Fix $k \geq 2$. There is a polynomial-time algorithm which takes as input a finite algebra $\mathbf{A}$ and returns the operation table of a k-ary cyclic term operation of $\mathbf{A}$ whenever such a term exists (and otherwise returns NO).*

Proof. Let $\mathbf{A}$ be a finite algebra with $|A| = N$ and fix an enumeration $\{\overline{a}_1, \dots, \overline{a}_{N^k}\} = A^k$. For a tuple $\overline{b} = (b_1, \dots, b_k) \in A^k$ and $i \in \{0, 1, \dots, k-1\}$ we define $\overline{b}^{(i)} := (b_{i+1}, b_{i+2}, \dots, b_k, b_1, b_2, \dots, b_i)$, the tuple obtained by cyclically permuting the coordinates of $\overline{b}$ a total of i times. The algorithm will start by constructing the operation tables of term operations of $\mathbf{A}$ which satisfy local equalities based on the equations satisfied by a cyclic term. Specifically, for each $\overline{a}_j \in A^k$ we build the operation table of a term operation $t_j(x_1, \dots, x_k)$ of $\mathbf{A}$ which satisfies:

$$t_j(\overline{a}_j) = t_j(\overline{a}_j^{(1)}) = \cdots = t_j(\overline{a}_j^{(k-1)}). \tag{4.1}$$

This can be achieved by generating the subalgebra $\left\langle \overline{a}_j,\ \overline{a}_j^{(1)},\ \dots,\ \overline{a}_j^{(k-1)} \right\rangle_{\mathbf{A}^k}$ and storing the operation table of any term operation of $\mathbf{A}$ which generates an element matching the equality pattern of $(x, x, \dots, x)$.[7] Thanks to Proposition 4.1.1 (and the discussion following that proposition) this takes time $O\left(||\mathbf{A}||^{2k+2}\right)$. Repeating this procedure for each $j = 1, \dots, N^k$ we see that the first stage of our algorithm takes time $O\left(N^k||\mathbf{A}||^{2k+2}\right)$ which is $O\left(||\mathbf{A}||^{3k+2}\right)$. If for some $1 \leq j \leq N^k$ we find that there is no such term operation $t_j(x_1, \dots, x_k)$, then clearly $\mathbf{A}$ does not have a cyclic term of arity k. In that case the algorithm will return the answer NO. Otherwise, thanks to [4, Lemma 2.4] (see also the rest of this proof), it follows that $\mathbf{A}$ does have a global cyclic term.

At this stage we have built the operation tables of term operations $t_j(x_1, \dots, x_k)$ for $1 \leq j \leq N^k$ which each satisfy

$$t_j(\overline{a}_j) = t_j(\overline{a}_j^{(1)}) = \cdots = t_j(\overline{a}_j^{(k-1)}).$$

We now inductively define for each $1 \leq n \leq N^k$ a term $t^n(x_1, \dots, x_k)$ which satisfies:

$$t^n(\overline{a}_i) = t^n(\overline{a}_i^{(1)}) = \cdots = t^n(\overline{a}_i^{(k-1)})$$

for every $1 \leq i \leq n$. The term t^{N^k} is therefore a cyclic term of $\mathbf{A}$.

The term $t^n(x_1, \dots, x_k)$ is defined via:

- $t^1(x_1, \dots, x_k) := t_1(x_1, \dots, x_k)$
- $t^{n+1}(x_1, \dots, x_k) :=$

  $t_u\left(t^n(x_1, x_2, \dots, x_{k-1}, x_k),\ t^n(x_2, \dots, x_{k-1}, x_k, x_1),\ \dots,\ t^n(x_k, x_1, x_2, \dots, x_{k-1})\right)$

[7] The Equalities 4.1 follow because the matrix whose columns are $\overline{a}_j, \overline{a}_j^{(1)}, \dots, \overline{a}_j^{(k-1)}$ is a symmetric matrix.

where u is chosen such that $\overline{a}_u = \left(t^n(\overline{a}_{n+1}), t^n(\overline{a}_{n+1}^{(1)}), \ldots, t^n(\overline{a}_{n+1}^{(k-1)})\right)$.

We first show that t^n satisfies the desired equalities. Notice that $t^1(x_1, \ldots, x_k)$ satisfies the desired $k-1$ equalities by the definition of $t_1(x_1, \ldots, x_k)$. Now suppose that $t^n(x_1, \ldots, x_k)$ satisfies the desired n sets of $k-1$ equalities and we show that $t^{n+1}(x_1, \ldots, x_k)$ satisfies these as well as the $(n+1)$-st.

Firstly, for $i \leq n$ and $0 \leq m \leq k-1$ we have:

$$\begin{aligned} t^{n+1}(\overline{a}_i^{(m)}) &= t_u(t^n(\overline{a}_i^{(m)}), t^n(\overline{a}_i^{(m+1)}), \ldots, t^n(\overline{a}_i^{(m+k-1)})) \\ &= t_u(t^n(\overline{a}_i), t^n(\overline{a}_i), \ldots, t^n(\overline{a}_i)) \end{aligned}$$

where addition in superscripts is performed modulo k and the second equality follows from the induction hypothesis on $t^n(x_1, \ldots, x_k)$. Since this value is independent of m, we see that $t^{n+1}(x_1, \ldots, x_k)$ satisfies the first n sets of $k-1$ equalities.

We also have

$$\begin{aligned} t^{n+1}(\overline{a}_{n+1}^{(m)}) &= t_u(t^n(\overline{a}_{n+1}^{(m)}), t^n(\overline{a}_{n+1}^{(m+1)}), \ldots, t^n(\overline{a}_{n+1}^{(m+k-1)})) \\ &= t_u(t^n(\overline{a}_{n+1}), t^n(\overline{a}_{n+1}^{(1)}), \ldots, t^n(\overline{a}_{n+1}^{(k-1)})) \end{aligned}$$

by the choice of u and the observation that

$$\left(t^n(\overline{a}_{n+1}^{(m)}),\ t^n(\overline{a}_{n+1}^{(m+1)}),\ \ldots,\ t^n(\overline{a}_{n+1}^{(m+k-1)})\right) = \left(t^n(\overline{a}_{n+1}),\ t^n(\overline{a}_{n+1}^{(1)}),\ \ldots,\ t^n(\overline{a}_{n+1}^{(k-1)})\right)^{(m)}.$$

Since the value of $t^{n+1}(\overline{a}_{n+1}^{(m)})$ is also independent of m, we see that $t^{n+1}(x_1, \ldots, x_k)$ also satisfies the final $k-1$ equalities, as required.

In this second stage of the algorithm, we have constructed the tables of N^k-many term operations, each of which can be obtained in time $O\left((k+1)N^k\right)$ using the previously constructed and stored tables. Since $(k+1)$ is a constant independent of the size of the instance $\mathbf{A}$, the runtime of the second stage of the algorithm is $O(N^k N^k)$ which is clearly $O(||\mathbf{A}||^{2k})$.

Hence the total runtime to obtain the table of a k-ary cyclic term operation of $\mathbf{A}$ is

$$O\left(||\mathbf{A}||^{3k+2}\right) + O\left(||\mathbf{A}||^{2k}\right) = O\left(||\mathbf{A}||^{3k+2}\right),$$

a polynomial in $||\mathbf{A}||$ as required. □

In the next section, we introduce a result first proved in [29, Section 4] which inspired the above result and all subsequent results found in this chapter.

4.3 Building a Mal'tsev Operation

Recall from Example 1.2.6: a *Mal'tsev term* is a term p satisfying the two equations $p(x, y, y) \approx x$ and $p(y, y, x) \approx x$. It was shown in [20, 24, 30, 42] that there are

polynomial-time algorithms to decide whether or not a finite idempotent algebra has a Mal'tsev term. The following stronger result was first demonstrated in [29]. We reproduce a proof of that result here to serve as a guide to understanding the proof of Theorem 4.4.5 in the next section, which uses the same technique in greater generality.

The result outlined in [29] is presented first as a result on obtaining a circuit for a Mal'tsev term operation of a given finite algebra $\mathbf{A}$, and then the operation table of the term operation is obtained as a corollary of the circuit result. As stated in Section 4.1, the algorithms outlined in this chapter all take the direct approach of building the operation tables of the relevant term operations. See the discussion following Proposition 4.1.1 for an analysis of why these are essentially the same problems. The proof below is adapted from the proof of [29, Theorem 6].

Theorem 4.3.1 ([29], Corollary 7)**.** *There is a polynomial-time algorithm which takes as input a finite idempotent algebra* $\mathbf{A}$ *and returns the operation table of some Mal'tsev term operation of* $\mathbf{A}$ *whenever such a term exists (and otherwise returns NO).*

Proof. Let $\mathbf{A}$ be a finite algebra with $|A| = N$ and fix an enumeration

$$\{(a_1, b_1), \dots, (a_{N^2}, b_{N^2})\} = A^2.$$

First we check that for each two-element subset $\{((a,b),1),((c,d),2)\} \subseteq A^2 \times [2]$ there is a term $t_{a,b,c,d}(x,y,z)$ of $\mathbf{A}$ satisfying $t_{a,b,c,d}(a,b,b) = a$ and $t_{a,b,c,d}(c,c,d) = d$. If for some choice of $a, b, c, d \in A$ there is no such term $t_{a,b,c,d}$, then clearly $\mathbf{A}$ has no Mal'tsev term, so the algorithm halts. Otherwise, we may store the operation table for each $t_{a,b,c,d}$ using the result of Proposition 4.1.1 by generating the subalgebra

$$\left\langle \begin{pmatrix} a \\ c \end{pmatrix}, \begin{pmatrix} b \\ c \end{pmatrix}, \begin{pmatrix} b \\ d \end{pmatrix} \right\rangle_{\mathbf{A}^2}$$

and halting when we find the tuple $\begin{pmatrix} a \\ d \end{pmatrix}$. Such a procedure can be completed in time $O(||\mathbf{A}||^7)$ using the result of Proposition 4.1.1.

There are N^4 such operation tables which gives a total run time for the first stage of this algorithm of $O(N^4||\mathbf{A}||^7)$ which is $O(||\mathbf{A}||^{11})$.

Next, for each $a, b \in A$, we inductively produce the operation table for a term $t_{a,b}(x,y,z)$ of $\mathbf{A}$ which satisfies: $t_{a,b}(a,b,b) = a$ and $t_{a,b}(x,x,y) = y$ for all $x, y \in A$. We define the term operation $t^j_{a,b}(x,y,z)$ for $1 \leq j < N^2$ as follows:

- $t^1_{a,b}(x,y,z) := t_{a,b,a_1,b_1}(x,y,z)$, and
- $t^{j+1}_{a,b}(x,y,z) := t_{a,b,u,b_{j+1}}(t^j_{a,b}(x,y,z), t^j_{a,b}(y,y,z), z)$ where $u = t^j_{a,b}(a_{j+1}, a_{j+1}, b_{j+1})$.

We claim that for each $1 \leq j \leq N^2$, the term $t^j_{a,b}(x,y,z)$ satisfies $t^j_{a,b}(a,b,b) = a$ and $t^j_{a,b}(a_k, a_k, b_k) = b_k$ whenever $1 \leq k \leq j \leq N^2$.

Note that $t^1_{a,b}(x,y,z)$ satisfies the desired equations by the definition of the term $t_{a,b,a_1,b_1}(x,y,z)$. Assume inductively that $t^j_{a,b}(x,y,z)$ satisfies the desired equation for some $1 \leq j < N^2$. We have:

- $t^{j+1}_{a,b}(a,b,b) = t_{a,b,u,b_{j+1}}(t^j_{a,b}(a,b,b), t^j_{a,b}(b,b,b), b) = t_{a,b,u,b_{j+1}}(a,b,b) = a$ as required, using the inductive hypothesis for $t^j_{a,b}(x,y,z)$, idempotency of $\mathbf{A}$, and the equations satisfied by $t_{a,b,u,b_{j+1}}$;
- If $1 \leq k < j+1$, then

$$\begin{aligned} t^{j+1}_{a,b}(a_k, a_k, b_k) &= t_{a,b,u,b_{j+1}}(t^j_{a,b}(a_k,a_k,b_k), t^j_{a,b}(a_k,a_k,b_k), b_k) \\ &= t_{a,b,u,b_{j+1}}(b_k,b_k,b_k) = b_k \end{aligned}$$

  as required, using the inductive hypothesis for $t^j_{a,b}(x,y,z)$ and idempotency of $\mathbf{A}$; and
- $t^{j+1}_{a,b}(a_{j+1}, a_{j+1}, b_{j+1}) = t_{a,b,u,b_{j+1}}(t^j_{a,b}(a_{j+1},a_{j+1},b_{j+1}), t^j_{a,b}(a_{j+1},a_{j+1},b_{j+1}), b_{j+1}) = b_{j+1}$ by the choice of u and definition of $t_{a,b,u,b_{j+1}}(x,y,z)$.

Choose $t_{a,b}(x,y,z) := t^{N^2}_{a,b}(x,y,z)$ and we have $t_{a,b}(a,b,b) = a$ and $t_{a,b}(x,x,y) = y$ for all $x, y \in A$ as required.

Constructing the table of $t^{j+1}_{a,b}(x,y,z)$ can clearly be achieved in time $O(N^3)$ using the (already stored) tables of $t^j_{a,b}(x,y,z)$ and $t_{a,b,u,b_{j+1}}(x,y,z)$. We repeat this construction N^2-many times to obtain $t_{a,b}(x,y,z)$, and we do this for every choice of $a, b \in A$. Hence the second step of this algorithm requires time $O(N^7)$ and hence also $O(||\mathbf{A}||^7)$.

The final stage of this algorithm is to inductively construct the table of a term $t_j(x,y,z)$ which satisfies $t_j(x,x,y) = y$ for all $x, y \in A$ and $t_j(a_k, b_k, b_k) = a_k$ for each $1 \leq k \leq j \leq N^2$. For $1 \leq j < N^2$ we define:

- $t_1(x,y,z) := t_{a_1,b_1}(x,y,z)$, and
- $t_{j+1}(x,y,z) := t_{a_{j+1},v}(x, t_j(x,y,y), t_j(x,y,z))$, where $v = t_j(a_{j+1}, b_{j+1}, b_{j+1})$

As above, it is easy to verify that $t_j(x,y,z)$ satisfies the desired equations for each $1 \leq j \leq N^2$, and hence $t_{N^2}(x,y,z)$ is a Mal'tsev term of $\mathbf{A}$ as required.

We see that the table of $t_{j+1}(x,y,z)$ can be constructed in time $O(N^3)$ using the previously stored tables of $t_j(x,y,z)$ and $t_{a_{j+1},v}(x,y,z)$. We repeat the construction N^2-many times to obtain $t_{N^2}(x,y,z)$, and hence the running time for this final stage of the algorithm is $O(N^5)$ which is $O(||\mathbf{A}||^5)$.

Hence the total running time of the algorithm described is

$$O(||\mathbf{A}||^{11}) + O(||\mathbf{A}||^7) + O(||\mathbf{A}||^5) = O(||\mathbf{A}||^{11}),$$

a polynomial in $||\mathbf{A}||$, as required. □

4.4 Building Term Operations using the Downward Column Condition

In this section we extend the result of [29] outlined in the previous section (Theorem 4.3.1) to Mal'tsev conditions which satisfy a certain syntactic condition. The *downward column condition* is a condition introduced by Horowitz in [24, Section 2] and is satisfied by the matrix defining the Mal'tsev condition of having a Mal'tsev term. Crucially, it is also satisfied by the matrix defining a k-edge term (for any fixed $k \in \mathbb{N}$), which will prove useful in Chapter 5. We begin by defining *strong E-terms* for a matrix E, before specifying the downward column condition and proving the generalization of Theorem 4.3.1.

Definition 4.4.1. ([24, Definition 2.2]) Let $E \in M_{m\times n}(\{x, y\})$ be a matrix and $t : A^n \to A$ an idempotent operation on the set A. We say that t is a *strong E-operation (on A)* if for every $1 \leq j \leq m$ we have

$$t(E_j(a, b)) = a \text{ for all } a, b \in A,$$

where $E_j(a, b)$ is the tuple obtained by substituting a for x and b for y in the j-th row of E. A term t of an algebra $\mathbf{A}$ is a *strong E-term (of $\mathbf{A}$)* if the term operation $t^{\mathbf{A}}(x_1, \ldots, x_n)$ is a strong E-operation on the underlying set A.

Example 4.4.2.

- A Mal'tsev term is a strong $\begin{pmatrix} x & y & y \\ y & y & x \end{pmatrix}$-term;
- A k-edge term is a strong E_k-term for the $(k \times (k+1))$-matrix

$$E_k = \begin{pmatrix} y & y & x & x & \ldots & x & x \\ y & x & y & x & \ldots & x & x \\ x & x & x & y & \ldots & x & x \\ \vdots & \vdots & \vdots & & \ddots & \vdots & \vdots \\ x & x & x & x & \ldots & y & x \\ x & x & x & x & \ldots & x & y \end{pmatrix};$$

- An n-ary near unanimity term is a strong U_n-term for the $(n \times n)$-matrix

$$U_n = \begin{pmatrix} y & x & x & \ldots & x & x \\ x & y & x & \ldots & x & x \\ x & x & y & \ldots & x & x \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x & x & x & \ldots & y & x \\ x & x & x & \ldots & x & y \end{pmatrix};$$

- A minority term is a strong $\begin{pmatrix} x & y & y \\ y & x & y \\ y & y & x \end{pmatrix}$-term.

Definition 4.4.3. ([24, Definition 2.5]) A matrix $E \in M_{m\times n}(\{x, y\})$ satisfies the *downward column condition (DCC)* if the columns of E, E^*, together with the column vector of all x's, $(x, x, \ldots, x)^T$, form a downward closed set in the lattice $\{x, y\}^m$ (where $x \leq y$). Equivalently, E satisfies the downward column condition iff whenever E^j is a column of E with more than one y, changing any y in the column E^j for an x produces another column E^i of the matrix E.

Example 4.4.4.

- The matrix $\begin{pmatrix} x & y & y \\ y & y & x \end{pmatrix}$, defining a Mal'tsev term, satisfies the DCC;
- The $(k \times (k+1))$-matrix

$$E_k = \begin{pmatrix} y & y & x & x & \ldots & x & x \\ y & x & y & x & \ldots & x & x \\ x & x & x & y & \ldots & x & x \\ \vdots & \vdots & \vdots & & \ddots & \vdots & \vdots \\ x & x & x & x & \ldots & y & x \\ x & x & x & x & \ldots & x & y \end{pmatrix}$$

  defining a k-edge term, satisfies the DCC;
- The $(n \times n)$-matrix

$$U_n = \begin{pmatrix} y & x & x & \ldots & x & x \\ x & y & x & \ldots & x & x \\ x & x & y & \ldots & x & x \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x & x & x & \ldots & y & x \\ x & x & x & \ldots & x & y \end{pmatrix}$$

  defining an n-ary near unanimity term, satisfies the DCC;
- The matrix $\begin{pmatrix} x & y & y \\ y & x & y \\ y & y & x \end{pmatrix}$, defining a minority term, does not satisfy the DCC, since (for example) the column $\begin{pmatrix} x \\ y \\ y \end{pmatrix}$ appears in the matrix but the column $\begin{pmatrix} x \\ x \\ y \end{pmatrix}$ does not.

Horowitz used this syntactical property of certain (matrices defining) Mal'tsev conditions to produce a general polynomial-time algorithm to determine the satisfaction by a finite idempotent algebra of any strong E-term condition where E satisfies the DCC [24, Theorem 2.6]. Analogously, we are able now to extend the result of Kazda, Opršal, Valeriote and Zhuk ([29, Corollary 7] and reproduced in the previous section as Theorem 4.3.1) to this broader class of conditions.

Theorem 4.4.5. *Let $E \in M_{m\times n}(\{x,y\})$ be a matrix over $\{x,y\}$ which satisfies the downward column condition (DCC). Then there is a polynomial-time algorithm which takes as input a finite idempotent algebra $\mathbf{A}$ and returns the operation table of some strong E-term operation of $\mathbf{A}$ whenever such a strong E-term exists (and otherwise returns NO).*

Proof. Firstly we note that thanks to [24, Theorem 2.6] we know that there is an algorithm to determine whether or not $\mathbf{A}$ has a strong E-term. We adapt the algorithm given in that paper to produce an operation table for such a term in the case that such a term exists.

Suppose $\mathbf{A}$ is finite and idempotent with $|A| = N$, and fix an enumeration

$$\{(a_1,b_1),\dots,(a_{N^2},b_{N^2})\} = A^2.$$

We begin by checking that for each m-element subset

$$\{((u_1,v_1),1),\dots,((u_m,v_m),m)\} \subseteq A^2 \times [m]$$

there is a term $t_{\overline{u},\overline{v}}(x_1,\dots,x_n)$ which satisfies $t_{\overline{u},\overline{v}}(E_j(u_j,v_j)) = u_j$ for each $1 \leq j \leq m$ (where $\overline{u} = (u_1,\dots,u_m), \overline{v} = (v_1,\dots,v_m)$). If for some subset we find that there is no such term, then the algorithm halts and outputs NO ($\mathbf{A}$ does not have a strong E-term). Otherwise, we store the operation tables of each term operation $t_{\overline{u},\overline{v}}$ found. Using the result of Proposition 4.1.1 this can be achieved in time $O(N^{2m}||\mathbf{A}||^{m+n+2})$ by generating each subalgebra

$$\left\langle \begin{pmatrix} E_1^1(u_1,v_1) \\ E_2^1(u_2,v_2) \\ \vdots \\ E_m^1(u_m,v_m) \end{pmatrix}, \begin{pmatrix} E_1^2(u_1,v_1) \\ E_2^2(u_2,v_2) \\ \vdots \\ E_m^2(u_m,v_m) \end{pmatrix}, \dots, \begin{pmatrix} E_1^n(u_1,v_1) \\ E_2^n(u_2,v_2) \\ \vdots \\ E_m^n(u_m,v_m) \end{pmatrix} \right\rangle_{\mathbf{A}^m}$$

and storing $t_{\overline{u},\overline{v}}$ as the operation giving rise to the tuple $\overline{u}^T$.

The proof will now proceed by inductively constructing for $0 \leq i \leq m$ and for each $\overline{u} = (u_{i+1},\dots,u_m), \overline{v} = (v_{i+1},\dots,v_m) \in A^{m-i}$ the operation table of a term operation $t^i_{\overline{u},\overline{v}}(x_1,\dots,x_n)$ which satisfies all of the equalities $t^i_{\overline{u},\overline{v}}(E_j(u_j,v_j)) = u_j$ (for each $j = i+1,\dots,m$) as well as globally satisfying the equations $t^i_{\overline{u},\overline{v}}(E_l(x,y)) \approx x$ for $1 \leq l \leq i$. The term t^m is therefore a strong E-term.[8] The base case when $i = 0$ was established in the previous paragraph.

[8]When $i = m$ there is precisely one tuple in A^{m-i} (the empty tuple) and hence there is no subscript indicated on this term. The inductive procedure described in the remainder of the proof still works to build the term $t^m(x,y,z)$ from the terms $t^{m-1}_{u_m,v_m}(x,y,z)$ previously constructed.

Each step of the induction will require inductively constructing the table of a term operation $t^{i,k}_{\overline{u},\overline{v}}(x_1,\ldots,x_n)$ which will satisfy all of the equalities $t^{i,k}_{\overline{u},\overline{v}}(E_j(u_j,v_j)) = u_j$ $(j = i+1,\ldots,m)$ and will satisfy $t^{i,k}_{\overline{u},\overline{v}}(E_l(u,v)) = u$ whenever $1 \leq l < i$ and $(u,v) \in A^2$ and also for $l = i$ whenever $(u,v) \in \{(a_1,b_1),\ldots,(a_k,b_k)\}$. The term $t^{i,N^2}_{\overline{u},\overline{v}}$ is therefore a term $t^i_{\overline{u},\overline{v}}$ satisfying the conditions in the previous paragraph.

Suppose that for each $\overline{u}',\overline{v}' \in A^{m-(i-1)}$ the table of a term operation $t^{i-1}_{\overline{u}',\overline{v}'}$ has been built satisfying the conditions outlined above. I.e. $t^{i-1}_{\overline{u}',\overline{v}'}$ satisfies all of the equations corresponding to rows 1 up to $i-1$, as well as satisfying the equalities $t^{i-1}_{\overline{u}',\overline{v}'}(E_j(\overline{u}'|_j,\overline{v}'|_j)) = \overline{u}'|_j$ for all $i \leq j \leq m$. Fix $\overline{u} = (u_{i+1},\ldots,u_m), \overline{v} = (v_{i+1},\ldots,v_m) \in A^{m-i}$. We inductively construct the table for $t^i_{\overline{u},\overline{v}}$ which globally satisfies the equations given by rows 1 up to i , and satisfies all the equalities

$$t^i_{\overline{u},\overline{v}}(E_j(u_j,v_j)) = u_j$$

for each $i+1 \leq j \leq m$. We define:

- $t^{i,1}_{\overline{u},\overline{v}}(x_1,\ldots,x_n) := t^{i-1}_{(a_1,u_{i+1},\ldots,u_m),(b_1,v_{i+1},\ldots,v_m)}(x_1,\ldots,x_n)$
- $t^{i,k+1}_{\overline{u},\overline{v}}(x_1,\ldots,x_n) := t^{i-1}_{\overline{\omega},\overline{\zeta}}(z_1(x_1,\ldots,x_n),\ldots,z_n(x_1,\ldots,x_n))$

where $\overline{\omega} = (a_{k+1},u_{i+1},\ldots,u_m)$, $\overline{\zeta} = (t^{i,k}_{\overline{u},\overline{v}}(E_i(a_{k+1},b_{k+1})),v_{i+1},\ldots,v_m)$, and (following [24]) for each $\alpha = 1,\ldots,n$ we define:

$$z_\alpha(x_1,\ldots,x_n) := \begin{cases} x_\alpha & \text{if } E^\alpha_i = x \\ & (E^\alpha_i \text{ is the entry in column } \alpha, \text{ row } i \text{ of } E) \\ t^{i,k}_{\overline{u},\overline{v}}(x_1,\ldots,x_n) & \text{if } E^\alpha \text{ has exactly one } y, \text{ in row } i \\ t^{i,k}_{\overline{u},\overline{v}}(E_i(x_q,x_\alpha)) & \text{else} \end{cases}$$

and q is chosen such that E^α and E^q differ only in row i (where $E^\alpha_i = y$, $E^q_i = x$). Such a q can be chosen since E satisfies the DCC.

Note that $t^{i,1}_{\overline{u},\overline{v}}(x_1,\ldots,x_n)$ satisfies the necessary conditions because they are precisely the inductive hypothesis for $t^{i-1}_{(a_1,u_{i+1},\ldots,u_m),(b_1,v_{i+1},\ldots,v_m)}(x_1,\ldots,x_n)$.

Assume that the operation table of a term operation $t^{i,k}_{\overline{u},\overline{v}}(x_1,\ldots,x_n)$ has been constructed and satisfies the necessary conditions. We need to verify that $t^{i,k+1}_{\overline{u},\overline{v}}$ also satisfies the necessary conditions. Firstly, for $j = i+1,\ldots,m$ we have:

$$t^{i,k+1}_{\overline{u},\overline{v}}(E_j(u_j,v_j)) = t^{i-1}_{\overline{\omega},\overline{\zeta}}(z_1(E_j(u_j,v_j)),\ldots,z_n(E_j(u_j,v_j)))$$

and for each $\alpha \leq n$ we have:

- If $E^\alpha_i = x$, then $z_\alpha(E_j(u_j,v_j)) = E^\alpha_j(u_j,v_j)$

- If E^α has exactly one y, in row i, then $z_\alpha(E_j(u_j, v_j)) = t^{i,k}_{\overline{u},\overline{v}}(E_j(u_j, v_j)) = u_j = E^\alpha_j(u_j, v_j)$ using the inductive hypothesis for $t^{i,k}_{\overline{u},\overline{v}}$ and the fact that $j \neq i$ (so $E^\alpha_j = x$)
- Otherwise, $z_\alpha(E_j(u_j, v_j)) = t^{i,k}_{\overline{u},\overline{v}}(E_i(E^q_j(u_j, v_j), E^\alpha_j(u_j, v_j))) = E^\alpha_j(u_j, v_j)$ since $t^{i,k}_{\overline{u},\overline{v}}$ is idempotent and $E^q_j = E^\alpha_j$ (since $j \neq i$).

It follows that

$$t^{i,k+1}_{\overline{u},\overline{v}}(E_j(u_j, v_j)) = t^{(i-1)}_{\overline{\omega},\overline{\zeta}}(z_1(E_j(u_j, v_j)), \ldots, z_n(E_j(u_j, v_j))) = t^{(i-1)}_{\overline{\omega},\overline{\zeta}}(E_j(u_j, v_j)) = u_j$$

using the inductive hypothesis for $t^{(i-1)}_{\overline{\omega},\overline{\zeta}}$, by the choice of $\overline{\omega}$ and $\overline{\zeta}$, since $j \geq i + 1$. This shows that $t^{i,k+1}_{\overline{u},\overline{v}}$ satisfies the relevant equalities for $j = i + 1, \ldots, m$.

Let $1 \leq l < i$, $1 \leq s \leq N^2$. Next we verify that $t^{i,k+1}_{\overline{u},\overline{v}}(E_l(a_s, b_s)) = a_s$. In this case we have for each $\alpha \leq n$:

- If $E^\alpha_i = x$, then $z_\alpha(E_l(a_s, b_s)) = E^\alpha_l(a_s, b_s)$
- If E^α has exactly one y, in row i, then $z_\alpha(E_l(a_s, b_s)) = t^{i,k}_{\overline{u},\overline{v}}(E_l(a_s, b_s)) = a_s = E^\alpha_l(a_s, b_s)$ using the inductive hypothesis for $t^{i,k}_{\overline{u},\overline{v}}$ and the fact that $l < i$
- Otherwise, $z_\alpha(E_l(a_s, b_s)) = t^{i,k}_{\overline{u},\overline{v}}(E_i(E^q_l(a_s, b_s), E^\alpha_l(a_s, b_s))) = E^\alpha_l(a_s, b_s)$ by idempotence, since $E^q_l = E^\alpha_l$ (because $l \neq i$).

Thus

$$t^{i,k+1}_{\overline{u},\overline{v}}(E_l(a_s, b_s)) = t^{(i-1)}_{\overline{\omega},\overline{\zeta}}(E_l(a_s, b_s)) = a_s$$

by the inductive hypothesis for $t^{(i-1)}_{\overline{\omega},\overline{\zeta}}$ (since $l \leq i - 1$). This shows that $t^{i,k+1}_{\overline{u},\overline{v}}$ satisfies the first $i - 1$ equations globally, as required.

Next we verify for $1 \leq s \leq k$ that $t^{i,k+1}_{\overline{u},\overline{v}}(E_i(a_s, b_s)) = a_s$ (which will leave us with only the same equation to verify for a_{k+1}, b_{k+1}). For $s < k + 1$ we have the following:

- If $E^\alpha_i = x$, then $z_\alpha(E_i(a_s, b_s)) = E^\alpha_i(a_s, b_s) = a_s$
- If E^α has exactly one y, in row i, then $z_\alpha(E_i(a_s, b_s)) = t^{i,k}_{\overline{u},\overline{v}}(E_i(a_s, b_s)) = a_s$ using the inductive hypothesis for $t^{i,k}_{\overline{u},\overline{v}}$ (since $s \leq k$)
- Otherwise, $z_\alpha(E_i(a_s, b_s)) = t^{i,k}_{\overline{u},\overline{v}}(E_i(E^q_i(a_s, b_s), E^\alpha_i(a_s, b_s))) = t^{i,k}_{\overline{u},\overline{v}}(E_i(a_s, b_s)) = a_s$ by the inductive hypothesis for $t^{i,k}_{\overline{u},\overline{v}}$ and the fact that $E^q_i = x$ and $E^\alpha_i = y$.

It follows that

$$t^{i,k+1}_{\overline{u},\overline{v}}(E_i(a_s, b_s)) = t^{(i-1)}_{\overline{\omega},\overline{\zeta}}(a_s, a_s, \ldots, a_s) = a_s$$

by the idempotency of $\mathbf{A}$.

Finally, we verify that $t^{i,k+1}_{\overline{u},\overline{v}}(E_i(a_{k+1}, b_{k+1})) = a_{k+1}$. In this case, we have:

- If $E_i^\alpha = x$, then $z_\alpha(E_i(a_{k+1}, b_{k+1})) = E_i^\alpha(a_{k+1}, b_{k+1}) = a_{k+1}$
- If E^α has exactly one y, in row i, then

$$z_\alpha(E_i(a_{k+1}, b_{k+1})) = t^{i,k}_{\overline{u},\overline{v}}(E_i(a_{k+1}, b_{k+1}))$$

- Otherwise,

$$\begin{aligned} z_\alpha(E_i(a_{k+1}, b_{k+1})) &= t^{i,k}_{\overline{u},\overline{v}}(E_i(E_i^q(a_{k+1}, b_{k+1}), E_i^\alpha(a_{k+1}, b_{k+1}))) \\ &= t^{i,k}_{\overline{u},\overline{v}}(E_i(a_{k+1}, b_{k+1})) \end{aligned}$$

since $E_i^q = x$ and $E_i^\alpha = y$.

Hence

$$t^{i,k+1}_{\overline{u},\overline{v}}(E_i(a_{k+1}, b_{k+1})) = t^{(i-1)}_{\overline{\omega},\overline{\zeta}}(E_i(a_{k+1}, t^{i,k}_{\overline{u},\overline{v}}(E_i(a_{k+1}, b_{k+1})))) = a_{k+1}$$

by the choice of $\overline{\omega}$ and $\overline{\zeta}$ and the inductive hypothesis for $t^{(i-1)}_{\overline{\omega},\overline{\zeta}}$.

It follows by induction that we can build the term $t^{i,N^2}_{\overline{u},\overline{v}}$ which satisfies all the equations necessary for $t^i_{\overline{u},\overline{v}}$ as outlined above. This completes the induction step for i, so we can inductively build the table of a strong E-term operation as required.

For fixed $i \in \{1, \ldots, m\}$ and fixed $\overline{u}, \overline{v} \in A^{m-i}$, the induction over k involves building the tables for N^2 many term operations, each of which can clearly be constructed in time $O(N^n)$ using previously stored operation tables.[9] There are $N^{2(m-i)}$ many such pairs of tuples $\overline{u}$ and $\overline{v}$ and so for fixed $i \in \{1, \ldots, m\}$ the induction process over k takes time $O(N^{2(m-i)}N^2N^n)$ which gives a total time requirement for the inductive procedure outlined of $O(N^{2m+n+2})$.

The total runtime of the algorithm is therefore

$$O(N^{2m}||\mathbf{A}||^{m+n+2}) + O(N^{2m+n+2}) = O(||\mathbf{A}||^{3m+n+2}).$$

□

The corollaries that follow mirror and extend those of [24, Section 2] to the problem of building operation tables for term operations satisfying specific strong Mal'tsev conditions.

Corollary 4.4.6. *For fixed $n > 2$ there is a polynomial-time algorithm which takes as input a finite idempotent algebra $\mathbf{A}$ and returns the operation table for an n-ary near unanimity term operation whenever $\mathbf{A}$ supports an n-ary near unanimity term.*

[9] For each tuple in A^n we need to look up at most $n+1$ values from previously stored operation tables. Each table lookup is a constant time procedure and $n+1$ is a fixed constant independent of the input size $||\mathbf{A}||$.

Proof. Since an n-ary near unanimity term is a strong U_n term for the matrix U_n defined in Example 4.4.2, Theorem 4.4.5 gives an algorithm whose runtime is $O(||\mathbf{A}||^{4n+2})$. □

Corollary 4.4.7. *For fixed $k \geq 2$ there is a polynomial-time algorithm which takes as input a finite idempotent algebra $\mathbf{A}$ and returns the operation table for a k-edge term operation whenever $\mathbf{A}$ supports a k-edge term.*

Proof. Since a k-edge term is a strong E_k-term for the matrix E_k defined in Example 4.4.2, Theorem 4.4.5 gives an algorithm whose runtime is $O(||\mathbf{A}||^{4k+3})$. □

In [24, Section 2] Horowitz also defines a property of (matrices defining) certain Mal'tsev conditions called the "local-global property". It is shown in [24] that if the matrix M has the local-global property then the idempotent satisfaction problem for the associated Mal'tsev condition is in P. To finish this section we define the local-global property more generally for Mal'tsev conditions. We will then see that Horowitz's result immediately extends to include all strong Mal'tsev conditions which have the local-global property and we will state a question about the related search problem of building term operations.

Definition 4.4.8. Let Σ be a strong Mal'tsev condition and $k \geq 1$. We say that Σ *has the (idempotent) local-global property of size k* if for any (idempotent) algebra $\mathbf{A}$ we have that $\mathbf{A}$ satisfies Σ if and only if for each k-element subset $S \subseteq A$ there are terms of $\mathbf{A}$ which satisfy the equations of Σ when restricted to the subset S.

We say that Σ *has the (idempotent) local-global property* if Σ has the (idempotent) local-global property of size k for some $k \geq 1$.

Proposition 4.4.9. *Let Σ be a strong Mal'tsev condition and $k \geq 1$ such that Σ has the (idempotent) local-global property of size k. Then the decision problem Sat_Σ (Sat^{Id}_Σ) is in P.*

Proof. As in [24, Lemma 2.4] we can design an algorithm to check that the equations of Σ can be "locally satisfied" by terms of $\mathbf{A}$ on subsets of size k by generating appropriate subpowers. The size of each of these subpowers is polynomial in $||\mathbf{A}||$ (although exponential in k and in the arities of the operation symbols appearing in Σ) and there are only polynomially many subsets of size k. Using [20, Proposition 6.1] this algorithm can be implemented in polynomial-time, as required. □

We now list some examples of conditions which have the local-global property.

Example 4.4.10.

- The Mal'tsev condition of having a Mal'tsev term has the idempotent local-global property of size 4 (follows from [24, Theorem 2.6]).
- Existence of an n-ary near unanimity term has the idempotent local-global property of size $2n$ (follows from [24, Theorem 2.6]).

- Existence of a k-ary edge term has the idempotent local-global property of size $2k$ (follows from [24, Theorem 2.6])
- Existence of a k-ary cyclic term has the local-global property of size k (follows from Theorem 4.2.1).
- Existence of a sequence of n Hagemann-Mitschke terms has the idempotent local-global property of size $2n + 2$ (follows from [42, Theorem 2.2]).
- Any strong, linear, idempotent Mal'tsev condition $M(P)$ arising from a pattern graph P via the construction outlined in [30, Subsection 3.2] has the idempotent local-global property (follows from [30, Theorem 7]).

The last item in the list of examples above is taken from [30]. In that paper Kazda and Valeriote construct graphs associated with various Mal'tsev conditions and show that satisfaction of the Mal'tsev condition in idempotent algebras is equivalent to the existence of a certain path within the constructed graph. We do not find it convenient to explore the details of that construction here but include the example as a suggestion for further study. In particular, the following conjecture naturally arises:

Conjecture 4.4.11. *Let Σ be a strong, linear, idempotent Mal'tsev condition arising as $M(P)$ for a pattern graph P as outlined in [30, Subsection 3.2]. There is a polynomial-time algorithm which takes as input a finite idempotent algebra $\mathbf{A}$ and returns term operations of $\mathbf{A}$ which satisfy the equations of Σ whenever such term operations exist.*

The results [29, Corollary 7] and Corollary 4.4.6 provide evidence supporting this conjecture since the Mal'tsev conditions "existence of a Mal'tsev term" and "existence of a majority term" are both demonstrated to arise as $M(P)$ for appropriate pattern graphs P in [30, Section 3]. Further evidence is provided by Theorem 4.5.1 in the next section since Hagemann-Mitschke terms are also shown in [30, Section 3] to fit into this framework. A proof of the conjecture would likely involve an induction similar to that found in the proof of Theorem 4.5.1. The following questions are also natural, although answers seem elusive.

Question 4.4.12. *Is there a strong Mal'tsev condition whose (idempotent) satisfaction problem is in P but which does not have the (idempotent) local-global property? [Is Minority an example?]*

Question 4.4.13. *Is there a strong Mal'tsev condition which has the local-global property but for which the corresponding search problem of obtaining term operations is not in P?*

Question 4.4.14. *Is there a strong Mal'tsev condition whose satisfaction problem is in P but for which the corresponding search problem of obtaining term operations is not in P?*

4.5 Building a Sequence of Hagemann-Mitschke Term Operations

In the previous sections we were able to construct term operations satisfying various Mal'tsev conditions each of which involved only a single operation symbol. In this section we highlight an algorithm similar to those of Theorems 4.2.1, 4.3.1, 4.4.5 for the Mal'tsev condition "Existence of a sequence of n Hagemann-Mitschke terms" defined in Example 1.2.11. The induction is a little more complicated because we are inductively defining a sequence of term operations rather than a single term operation but the method of proof is essentially the same as in the single term case. The algorithm is based on the proof of [42, Theorem 2.2].

Theorem 4.5.1. *Let $n \geq 1$. There is a polynomial-time algorithm which takes as input a finite idempotent algebra $\mathbf{A}$ and returns the operation tables of a sequence of n Hagemann-Mitschke term operations of $\mathbf{A}$ whenever such terms exist.*

Proof. Suppose $\mathbf{A}$ is a finite idempotent algebra with $|A| = N$, and fix an enumeration

$$\{(a_1, b_1), \ldots, (a_{N^2}, b_{N^2})\} = A^2.$$

We begin by checking that for each $(n+1)$-element sequence $((u_0, v_0), \ldots, (u_n, v_n))$ over A^2 there are terms $p_{0,\overline{u},\overline{v}}(x, y, z) := x$, $p_{j,\overline{u},\overline{v}}(x, y, z)$ for $1 \leq j \leq n$ and $p_{n+1}(x, y, z) := z$ which satisfy $p_{j,\overline{u},\overline{v}}(u_j, u_j, v_j) = p_{j+1,\overline{u},\overline{v}}(u_j, v_j, v_j)$ for each $0 \leq j \leq n$. If for some subset we find that there is no such sequence of terms, then the algorithm halts and outputs the answer that $\mathbf{A}$ does not have a sequence of n Hagemann-Mitschke terms (and hence does not generate a congruence $(n+1)$-permutable variety) using the result of [42, Theorem 2.2].

The procedure for deciding whether or not such terms exist is outlined in [42, Corollary 2.3] and we reproduce it here to see how the term operations themselves are obtained. Simply, we generate each of the subalgebras

$$R_j := \left\langle \begin{pmatrix} u_j \\ u_{j+1} \end{pmatrix}, \begin{pmatrix} v_j \\ u_{j+1} \end{pmatrix}, \begin{pmatrix} v_j \\ v_{j+1} \end{pmatrix} \right\rangle$$

for $j = 0, \ldots, n-1$ and search for elements $c_1, \ldots, c_{n-1} \in A$ such that $\begin{pmatrix} u_0 \\ c_1 \end{pmatrix} \in R_0$, $\begin{pmatrix} c_j \\ c_{j+1} \end{pmatrix} \in R_j$ for $j = 1, \ldots, n-2$ and $\begin{pmatrix} c_{n-1} \\ v_n \end{pmatrix} \in R_{n-1}$.

For fixed $\overline{u}, \overline{v}$ we generate n different subalgebras of A^2. Using the result of Proposition 4.1.1 we can generate these n subalgebras and store operation tables for term operations giving rise to each element in time $O(n||\mathbf{A}||^{3+2+2}) = O(||\mathbf{A}||^7)$. We then need to search for the elements $c_1, \ldots, c_{n-1}$. This can be achieved using standard graph-theoretic techniques[10] in time $O(nN)$. This gives a total time requirement for the

[10]Here we consider each R_j as a collection of edges between disjoint copies of A and ask "Is there a path from u_0 in the first copy to v_n in the last?"

first stage of this algorithm (looping over all $\overline{u}, \overline{v} \in A^{n+1}$) of $O((N^{n+1})^2(||\mathbf{A}||^7 + nN))$ which is $O(||\mathbf{A}||^{2n+9})$.

Next we will use induction to construct, for each $0 \leq i \leq n+1$ and all $\overline{u} := (u_i, \ldots, u_n)$, $\overline{v} := (v_i, \ldots, v_n) \in A^{n+1-i}$, the operation tables of a sequence of term operations

$$x \approx p^i_{0,\overline{u},\overline{v}}(x,y,z), p^i_{1,\overline{u},\overline{v}}(x,y,z), \ldots, p^i_{n,\overline{u},\overline{v}}(x,y,z), p^i_{n+1,\overline{u},\overline{v}}(x,y,z) \approx z$$

such that :

$$p^i_{m,\overline{u},\overline{v}}(x,x,y) \approx p^i_{m+1,\overline{u},\overline{v}}(x,y,y) \text{ if } 0 \leq m < i \tag{4.2}$$

and

$$p^i_{m,\overline{u},\overline{v}}(u_m,u_m,v_m) = p^i_{m+1,\overline{u},\overline{v}}(u_m,v_m,v_m) \text{ for each } i \leq m \leq n \tag{4.3}$$

The base of this induction (when $i = 0$) was already established in an earlier paragraph. For the inductive step, suppose that the sequence of terms

$$p^{i-1}_{1,\overline{u}',\overline{v}'}(x,y,z), \ldots, p^{i-1}_{n,\overline{u}',\overline{v}'}(x,y,z)$$

has been constructed for each $\overline{u}', \overline{v}' \in A^{n+1-(i-1)}$. For each $\overline{u} = (u_i, \ldots, u_n)$, $\overline{v} = (v_i, \ldots, v_n) \in A^{n+1-i}$ we inductively construct new sequences of terms

$$x \approx p^{i,k}_{0,\overline{u},\overline{v}}(x,y,z), p^{i,k}_{1,\overline{u},\overline{v}}(x,y,z), \ldots, p^{i,k}_{n,\overline{u},\overline{v}}(x,y,z), p^{i,k}_{n+1,\overline{u},\overline{v}}(x,y,z) \approx z$$

for $1 \leq k \leq N^2$ such that:

$$p^{i,k}_{m,\overline{u},\overline{v}}(x,x,y) \approx p^{i,k}_{m+1,\overline{u},\overline{v}}(x,y,y) \text{ if } 0 \leq m < i-1, \tag{4.4}$$

$$p^{i,k}_{i-1,\overline{u},\overline{v}}(u,u,v) = p^{i,k}_{i,\overline{u},\overline{v}}(u,v,v) \text{ if } (u,v) \in \{(a_1,b_1), \ldots, (a_k,b_k)\} \tag{4.5}$$

and

$$p^{i,k}_{m,\overline{u},\overline{v}}(u_m,u_m,v_m) = p^{i,k}_{m+1,\overline{u},\overline{v}}(u_m,v_m,v_m) \text{ for each } i \leq m \leq n \tag{4.6}$$

The definition of these terms is given by:

- For $1 \leq m \leq n$ define: $p^{i,1}_{m,\overline{u},\overline{v}}(x,y,z) := p^{i-1}_{m,(a_1,u_i,\ldots,u_n),(b_1,v_i,\ldots,v_n)}(x,y,z)$
- For $1 \leq k < N^2$:
  - For $1 \leq m < i-1$ define:

$$p^{i,k+1}_{m,\overline{u},\overline{v}}(x,y,z) := p^{i-1}_{m,\overline{\omega},\overline{\zeta}}(p^{i,k}_{m,\overline{u},\overline{v}}(x,y,z), p^{i,k}_{m,\overline{u},\overline{v}}(y,y,z), z)$$

  - Define:

$$p^{i,k+1}_{i-1,\overline{u},\overline{v}}(x,y,z) := p^{i-1}_{i-1,\overline{\omega},\overline{\zeta}}(p^{i,k}_{i-1,\overline{u},\overline{v}}(x,y,z), p^{i,k}_{i-1,\overline{u},\overline{v}}(y,y,z), p^{i,k}_{i,\overline{u},\overline{v}}(y,z,z))$$

– Define:

$$p^{i,k+1}_{i,\overline{u},\overline{v}}(x,y,z) := p^{i-1}_{i,\overline{\omega},\overline{\zeta}}(p^{i,k}_{i-1,\overline{u},\overline{v}}(x,x,y), p^{i,k}_{i,\overline{u},\overline{v}}(x,y,y), p^{i,k}_{i,\overline{u},\overline{v}}(x,y,z))$$

– For $i+1 \leq m \leq n$ define:

$$p^{i,k+1}_{m,\overline{u},\overline{v}}(x,y,z) := p^{i-1}_{m,\overline{\omega},\overline{\zeta}}(x, p^{i,k}_{m,\overline{u},\overline{v}}(x,y,y), p^{i,k}_{m,\overline{u},\overline{v}}(x,y,z))$$

where

$$\overline{\omega} := (p^{i,k}_{i-1,\overline{u},\overline{v}}(a_{k+1}, a_{k+1}, b_{k+1}), u_i, u_{i+1}, \ldots, u_{n-1}, u_n) \in A^{n+1-(i-1)}$$

and

$$\overline{\zeta} := (p^{i,k}_{i,\overline{u},\overline{v}}(a_{k+1}, b_{k+1}, b_{k+1}), p^{i,k}_{i,\overline{u},\overline{v}}(u_i, u_i, v_i), p^{i,k}_{i+1,\overline{u},\overline{v}}(u_{i+1}, u_{i+1}, v_{i+1}), \\ \ldots, p^{i,k}_{n-1,\overline{u},\overline{v}}(u_{n-1}, u_{n-1}, v_{n-1}), p^{i,k}_{n,\overline{u},\overline{v}}(u_n, u_n, v_n)) \in A^{n+1-(i-1)}$$

We first verify that these new sequences of terms satisfy the necessary equations (Equations 4.4, 4.5, and 4.6), beginning with the base case $k = 1$ and assuming the inductive hypothesis that Equations 4.2 and 4.3 hold with $i-1$ in place of i.

To verify 4.4 when $k = 1$, observe that for $m < i-1$ and $a, b \in A$ we have:

$$p^{i,1}_{m,\overline{u},\overline{v}}(a,a,b) = p^{i-1}_{m,(a_1,u_i,\ldots,u_n),(b_1,v_i,\ldots,v_n)}(a,a,b) \\ = p^{i-1}_{m+1,(a_1,u_i,\ldots,u_n),(b_1,v_i,\ldots,v_n)}(a,b,b) = p^{i,1}_{m+1,\overline{u},\overline{v}}(a,b,b)$$

where the innermost equality follows from the inductive hypothesis 4.2 since $m < i-1$.

To verify 4.5 when $k = 1$, observe that we have:

$$p^{i,1}_{i-1,\overline{u},\overline{v}}(a_1,a_1,b_1) = p^{i-1}_{i-1,(a_1,u_i,\ldots,u_n),(b_1,v_i,\ldots,v_n)}(a_1,a_1,b_1) \\ = p^{i-1}_{i,(a_1,u_i,\ldots,u_n),(b_1,v_i,\ldots,v_n)}(a_1,b_1,b_1) = p^{i,1}_{i,\overline{u},\overline{v}}(a_1,b_1,b_1)$$

where the innermost equality follows from the inductive hypothesis 4.3 with $m = i-1$.

To verify 4.6 when $k = 1$, observe that for $i \leq m \leq n$ and

$$(u,v) \in \{(u_i,v_i), \ldots, (u_n,v_n)\}$$

we have:

$$p^{i,1}_{m,\overline{u},\overline{v}}(u,u,v) = p^{i-1}_{m,(a_1,u_i,\ldots,u_n),(b_1,v_i,\ldots,v_n)}(u,u,v) \\ = p^{i-1}_{m+1,(a_1,u_i,\ldots,u_n),(b_1,v_i,\ldots,v_n)}(u,v,v) = p^{i,1}_{m+1,\overline{u},\overline{v}}(u,v,v)$$

where the innermost equality follows from the inductive hypothesis 4.3 since $i-1 \leq m \leq n$.

Now we assume (in addition to the inductive hypothesis for $i-1$) the additional inductive hypotheses of Equations 4.4, 4.5, and 4.6 for k. We need to verify these three equations for $k+1$.

To verify 4.4 for $k+1$, observe that for $0 \leq m < i-2$ and $a, b \in A$ we have:

$$\begin{aligned}
p^{i,k+1}_{m,\overline{u},\overline{v}}(a,a,b) &= p^{i-1}_{m,\overline{\omega},\overline{\zeta}}(p^{i,k}_{m,\overline{u},\overline{v}}(a,a,b), p^{i,k}_{m,\overline{u},\overline{v}}(a,a,b), b) \\
&= p^{i-1}_{m+1,\overline{\omega},\overline{\zeta}}(p^{i,k}_{m,\overline{u},\overline{v}}(a,a,b), b, b) \text{ using the inductive hypothesis for } i-1 \text{ (4.2)} \\
&= p^{i-1}_{m+1,\overline{\omega},\overline{\zeta}}(p^{i,k}_{m+1,\overline{u},\overline{v}}(a,b,b), b, b) \text{ using the inductive hypothesis for } k \text{ (4.4)} \\
&= p^{i-1}_{m+1,\overline{\omega},\overline{\zeta}}(p^{i,k}_{m+1,\overline{u},\overline{v}}(a,b,b), p^{i,k}_{m+1,\overline{u},\overline{v}}(b,b,b), b) \text{ using idempotency of } \mathbf{A} \\
&= p^{i,k+1}_{m+1,\overline{u},\overline{v}}(a,b,b)
\end{aligned}$$

noting that the uses of 4.2 and 4.4 are valid since $0 \leq m < i-2$.

When $m = i-2$ we have:

$$\begin{aligned}
p^{i,k+1}_{i-2,\overline{u},\overline{v}}(a,a,b) &= p^{i-1}_{i-2,\overline{\omega},\overline{\zeta}}(p^{i,k}_{i-2,\overline{u},\overline{v}}(a,a,b), p^{i,k}_{i-2,\overline{u},\overline{v}}(a,a,b), b) \\
&= p^{i-1}_{i-1,\overline{\omega},\overline{\zeta}}(p^{i,k}_{i-2,\overline{u},\overline{v}}(a,a,b), b, b) \text{ using the inductive hypothesis for } i-1 \text{ (4.2)} \\
&= p^{i-1}_{i-1,\overline{\omega},\overline{\zeta}}(p^{i,k}_{i-1,\overline{u},\overline{v}}(a,b,b), b, b) \text{ using the inductive hypothesis for } k \text{ (4.4)} \\
&= p^{i-1}_{i-1,\overline{\omega},\overline{\zeta}}(p^{i,k}_{i-1,\overline{u},\overline{v}}(a,b,b), p^{i,k}_{i-1,\overline{u},\overline{v}}(b,b,b), p^{i,k}_{i-1,\overline{u},\overline{v}}(b,b,b)) \text{ using idempotency of } \mathbf{A} \\
&= p^{i,k+1}_{i-1,\overline{u},\overline{v}}(a,b,b)
\end{aligned}$$

noting that the uses of 4.2 and 4.4 are valid since $i-2 < i-1$.

To verify that 4.5 holds for $k+1$, first let $(u,v) \in \{(a_1,b_1),\ldots,(a_k,b_k)\}$. We have:

$$\begin{aligned}
p^{i,k+1}_{i-1,\overline{u},\overline{v}}(u,u,v) &= p^{i-1}_{i-1,\overline{\omega},\overline{\zeta}}(p^{i,k}_{i-1,\overline{u},\overline{v}}(u,u,v), p^{i,k}_{i-1,\overline{u},\overline{v}}(u,u,v), p^{i,k}_{i,\overline{u},\overline{v}}(u,v,v)) \\
&= p^{i-1}_{i-1,\overline{\omega},\overline{\zeta}}(p^{i,k}_{i,\overline{u},\overline{v}}(u,v,v), p^{i,k}_{i,\overline{u},\overline{v}}(u,v,v), p^{i,k}_{i,\overline{u},\overline{v}}(u,v,v)) \\
&\quad \text{using the inductive hypothesis for } k \text{ (4.5)} \\
&= p^{i,k}_{i,\overline{u},\overline{v}}(u,v,v) \text{ using idempotency of } \mathbf{A} \\
&= p^{i-1}_{i,\overline{\omega},\overline{\zeta}}(p^{i,k}_{i,\overline{u},\overline{v}}(u,v,v), p^{i,k}_{i,\overline{u},\overline{v}}(u,v,v), p^{i,k}_{i,\overline{u},\overline{v}}(u,v,v)) \text{ using idempotency of } \mathbf{A} \\
&= p^{i-1}_{i,\overline{\omega},\overline{\zeta}}(p^{i,k}_{i-1,\overline{u},\overline{v}}(u,u,v), p^{i,k}_{i,\overline{u},\overline{v}}(u,v,v), p^{i,k}_{i,\overline{u},\overline{v}}(u,v,v)) \\
&\quad \text{using the inductive hypothesis for } k \text{ (4.5)} \\
&= p^{i,k+1}_{i,\overline{u},\overline{v}}(u,v,v)
\end{aligned}$$

where the use of 4.5 is valid (in both cases) since $(u,v) \in \{(a_1,b_1),\ldots,(a_k,b_k)\}$.

We also need to verify 4.5 for the pair (a_{k+1}, b_{k+1}). In this instance, we have:

$$\begin{aligned}
&p^{i,k+1}_{i-1,\overline{u},\overline{v}}(a_{k+1}, a_{k+1}, b_{k+1}) \\
&= p^{i-1}_{i-1,\overline{\omega},\overline{\zeta}}(p^{i,k}_{i-1,\overline{u},\overline{v}}(a_{k+1}, a_{k+1}, b_{k+1}), p^{i,k}_{i-1,\overline{u},\overline{v}}(a_{k+1}, a_{k+1}, b_{k+1}), p^{i,k}_{i,\overline{u},\overline{v}}(a_{k+1}, b_{k+1}, b_{k+1})) \\
&= p^{i-1}_{i,\overline{\omega},\overline{\zeta}}(p^{i,k}_{i-1,\overline{u},\overline{v}}(a_{k+1}, a_{k+1}, b_{k+1}), p^{i,k}_{i,\overline{u},\overline{v}}(a_{k+1}, b_{k+1}, b_{k+1}), p^{i,k}_{i,\overline{u},\overline{v}}(a_{k+1}, b_{k+1}, b_{k+1})) \\
&= p^{i,k+1}_{i,\overline{u},\overline{v}}(a_{k+1}, b_{k+1}, b_{k+1})
\end{aligned}$$

where the innermost equality holds by the inductive hypothesis for $i-1$ (4.3) and the choice of $\overline{\omega}, \overline{\zeta}$.

We next need to verify that 4.6 holds for $k+1$. We begin by verifying the equation for $m = i$ and then finally confirm the result for $i+1 \leq m \leq n$.

To verify 4.6 for $m = i$, we have:

$$\begin{aligned}
p^{i,k+1}_{i,\overline{u},\overline{v}}(u_i, u_i, v_i) &= p^{i-1}_{i,\overline{\omega},\overline{\zeta}}(p^{i,k}_{i-1,\overline{u},\overline{v}}(u_i, u_i, u_i), p^{i,k}_{i,\overline{u},\overline{v}}(u_i, u_i, u_i), p^{i,k}_{i,\overline{u},\overline{v}}(u_i, u_i, v_i)) \\
&= p^{i-1}_{i,\overline{\omega},\overline{\zeta}}(u_i, u_i, p^{i,k}_{i,\overline{u},\overline{v}}(u_i, u_i, v_i)) \quad \text{using idempotency of } \mathbf{A} \\
&= p^{i-1}_{i+1,\overline{\omega},\overline{\zeta}}(u_i, p^{i,k}_{i,\overline{u},\overline{v}}(u_i, u_i, v_i), p^{i,k}_{i,\overline{u},\overline{v}}(u_i, u_i, v_i)) \\
&\quad \text{using the inductive hypothesis for } i-1 \text{ (4.3 with } m = i) \\
&= p^{i-1}_{i+1,\overline{\omega},\overline{\zeta}}(u_i, p^{i,k}_{i+1,\overline{u},\overline{v}}(u_i, v_i, v_i), p^{i,k}_{i+1,\overline{u},\overline{v}}(u_i, v_i, v_i)) \\
&\quad \text{using the inductive hypothesis for } k \text{ (4.6 with } m = i) \\
&= p^{i,k+1}_{i+1,\overline{u},\overline{v}}(u_i, v_i, v_i)
\end{aligned}$$

where the use of 4.3 is valid by the choice of $\overline{\omega}, \overline{\zeta}$.

To verify 4.6 for $i+1 \leq m < n$ we have:

$$\begin{aligned}
p^{i,k+1}_{m,\overline{u},\overline{v}}(u_m, u_m, v_m) &= p^{i-1}_{m,\overline{\omega},\overline{\zeta}}(u_m, p^{i,k}_{m,\overline{u},\overline{v}}(u_m, u_m, u_m), p^{i,k}_{m,\overline{u},\overline{v}}(u_m, u_m, v_m)) \\
&= p^{i-1}_{m,\overline{\omega},\overline{\zeta}}(u_m, u_m, p^{i,k}_{m,\overline{u},\overline{v}}(u_m, u_m, v_m)) \quad \text{using idempotency of } \mathbf{A} \\
&= p^{i-1}_{m+1,\overline{\omega},\overline{\zeta}}(u_m, p^{i,k}_{m,\overline{u},\overline{v}}(u_m, u_m, v_m), p^{i,k}_{m,\overline{u},\overline{v}}(u_m, u_m, v_m)) \\
&\quad \text{using the inductive hypothesis for } i-1 \text{ (4.3)} \\
&= p^{i-1}_{m+1,\overline{\omega},\overline{\zeta}}(u_m, p^{i,k}_{m+1,\overline{u},\overline{v}}(u_m, v_m, v_m), p^{i,k}_{m+1,\overline{u},\overline{v}}(u_m, v_m, v_m)) \\
&\quad \text{using the inductive hypothesis for } k \text{ (4.6)} \\
&= p^{i,k+1}_{m+1,\overline{u},\overline{v}}(u_m, v_m, v_m)
\end{aligned}$$

where the use of 4.3 is valid since $i-1 \leq m \leq n$ and by the choice of $\overline{\omega}, \overline{\zeta}$, and 4.6 is valid since $i \leq m \leq n$.

We are finally ready to verify 4.6 for $m = n$. In this case, we have:

$$\begin{aligned} p^{i,k+1}_{n,\overline{u},\overline{v}}(u_n, u_n, v_n) &= p^{i-1}_{n,\overline{\omega},\overline{\zeta}}(u_n, p^{i,k}_{n,\overline{u},\overline{v}}(u_n, u_n, u_n), p^{i,k}_{n,\overline{u},\overline{v}}(u_n, u_n, v_n)) \\ &= p^{i-1}_{n,\overline{\omega},\overline{\zeta}}(u_n, u_n, p^{i,k}_{n,\overline{u},\overline{v}}(u_n, u_n, v_n)) \text{ using idempotency of } \mathbf{A} \\ &= p^{i,k}_{n,\overline{u},\overline{v}}(u_n, u_n, v_n) \text{ using the inductive hypothesis for } i-1 \text{ (4.3)} \\ &= v_n \text{ using the inductive hypothesis for } k \text{ (4.6)} \end{aligned}$$

where the use of 4.3 is valid since $i - 1 \leq m = n$ and by the choice of $\overline{\omega}, \overline{\zeta}$, and 4.6 is valid since $i \leq m = n$.

This ends the inductive argument for k+1. Choosing $p^{i}_{m,\overline{u},\overline{v}}(x, y, z) := p^{i,N^2}_{m,\overline{u},\overline{v}}(x, y, z)$ we see that Equations 4.2 and 4.3 are satisfied for this new term, ending the inductive argument for i. The term operations $p^n_m(x, y, z)$ clearly form a Hagemann-Mitschke sequence of term operations for $\mathbf{A}$, as required.

The only thing left to do is to analyze the complexity of this induction process. Given operation tables for the term operations $p^{i,k}_{m,\overline{u},\overline{v}}(x, y, z)$ and $p^{i-1}_{m,\overline{\omega},\overline{\zeta}}(x, y, z)$ we can build the operation tables for $p^{i,k+1}_{m,\overline{u},\overline{v}}(x, y, z)$ using at most four table look-ups (depending on the value of m) for each input tuple. Hence building $p^{i,k+1}_{m,\overline{u},\overline{v}}(x, y, z)$ takes time $O(N^3) = O(||\mathbf{A}||^3)$.[11] Repeating this for each $1 \leq m \leq n$ gives a total time to build the entire sequence $(p^{i,k+1}_{m,\overline{u},\overline{v}}(x, y, z))^n_{m=1}$ of $O(n||\mathbf{A}||^3)$. We repeat this procedure for each $1 \leq k \leq N^2$, requiring time $O(n||\mathbf{A}||^3(N^2)) = O(n||\mathbf{A}||^5)$, and for every $\overline{u}, \overline{v} \in A^{n+1-i}$. This gives a total time requirement for the inductive procedure of $O(\sum_{i=1}^{n} N^{n+1-i}(n||\mathbf{A}||^5)) = O(nN^n||\mathbf{A}||^5) = O(n||\mathbf{A}||^{n+5}) = O(||\mathbf{A}||^{n+5})$.

The total runtime of the algorithm to produce the entire sequence is therefore

$$O(||\mathbf{A}||^{2n+9}) + O(||\mathbf{A}||^{n+5}) = O(||\mathbf{A}||^{2n+9})$$

a polynomial in $||\mathbf{A}||$ as required. □

As mentioned in the previous section (Conjecture 4.4.11) this result provides some limited evidence that a similar algorithm could work for the broad range of Mal'tsev conditions described by paths in the sense of [30]. We underline this again as a potentially fruitful immediate extension of the work outlined in this thesis.

[11] We also need initially $(n - i + 3)$-many table look-ups to determine the tuples $\overline{\omega}, \overline{\zeta}$ which does not affect the asymptotic estimate obtained.

Chapter 5

An NP Result for Cube Terms

In this chapter, we use a recent result of Bulatov, Mayr, and Szendrei [13, Theorem 4.13] to show that the problem $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ is in NP whenever Σ is a strong Mal'tsev condition of height < 1. In particular, testing for the presence of any particular cube term[1] for a finite algebra $\mathbf{A}$ is in NP. We begin by showing that any nontrivial strong Mal'tsev condition of height < 1 implies the existence of a cube term. Although not always using this terminology, similar observations to those found in Section 5.2 are found in many other sources, notably in [8] (where the phrase cube term was first used but with a slightly different definition) and in [32].

From our perspective, one very useful feature of cube terms is that any algebra with a k-cube term of some arity also has a k-edge term (of arity $k+1$) and vice versa ([8, Theorem 2.12]). Testing for the presence of a k-edge term in an idempotent algebra is a polynomial-time decision problem ([24, Corollary 2.7(2)]) and using Corollary 4.4.7 we can even obtain the operation table of a k-edge term operation in polynomial-time. In this chapter we combine Corollary 4.4.7 with the result [32, Theorem 3.5] to obtain the operation table of a $(1, k-1)$-parallelogram term operation for the idempotent algebra $\mathbf{A}$ (whenever $\mathbf{A}$ supports such a term). This will allow us to use an NP-oracle defined in [13, Subsection 4.3] to verify YES instances of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ in polynomial-time which establishes the result $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma} \in \mathsf{NP}$. The proof relies on building "representations" (see Definition 5.2.12) for subalgebras which are particular generating sets in which every element of the generated subalgebra can be obtained efficiently using only repeated applications of the $(1, k-1)$-parallelogram term operation. More specifically, given a YES instance $\mathbf{A}$ of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$, the polynomial-time verification works as follows:

1. Build the operation table of a $(1, k-1)$-parallelogram term operation for $\mathbf{A}$;
2. Build an instance $g, \pi_1, \ldots, \pi_r$ of $\mathrm{SMP}(\mathbf{A})$ which is equivalent to the instance $\mathbf{A}$ of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$;
3. Use the NP-oracle defined in [13, Subsection 4.3] to build a "representation" for $\langle \pi_1, \ldots, \pi_r \rangle$;

[1] See Example 1.2.9 or Definition 5.1.3.

4. Verify that $g, \pi_1, \ldots, \pi_r$ is a YES instance of SMP($\mathbf{A}$) and hence that $\mathbf{A}$ is a YES instance of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$.

Indeed, in establishing item 2 outlined above what we actually achieve is a polynomial-time reduction of the problem $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ to the problem given by:

- INPUT: A finite algebra $\mathbf{A}$ with a $(1, k-1)$-parallelogram operation as a basic operation, and $\overline{a}, \overline{b}_1, \ldots, \overline{b}_r \in A^n$
- QUESTION: Is $\overline{a}$ in the subalgebra of $\mathbf{A}^n$ generated by $\{\overline{b}_1, \ldots, \overline{b}_r\}$?

A similar observation is found in the conclusion of [29] and we note analogously that any tractability result for the problem outlined above will also lead to a polynomial-time algorithm for the problem $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$.

In the next section we begin our discussion of the problem by outlining the connection between Mal'tsev conditions of height < 1 and cube terms.

5.1 Height < 1 Mal'tsev Conditions Imply Cube Terms

We begin by specifying what we mean by a Mal'tsev condition "of height < 1". The definition presented here is inspired by [6, Definition 5.1].

Definition 5.1.1. An equation is *of height* < 1 if it takes one of the forms:

- $f(x_1, \ldots, x_n) \approx x_i$
- $x_i \approx f(x_1, \ldots, x_n)$
- $x_i \approx x_j$

where $x_1, \ldots, x_n$ are (possibly repeated) variables and f is a single operation symbol.

The Mal'tsev condition Σ is *of height* < 1 if there is a Mal'tsev condition Γ which is equivalent to Σ (see Definition 1.2.1) and such that the equations in Γ are all of height < 1.[2]

Example 5.1.2.

- The equation $m(x, y, y) \approx x$ is of height < 1,
- The Mal'tsev condition of having a minority term

$$\text{Minority} := \{m(x, y, y) \approx x,\ m(y, x, y) \approx x,\ m(y, y, x) \approx x\}$$

is of height < 1,

[2]When working with a Mal'tsev condition of height < 1 we will assume from now on that the condition is presented as a set of equations of height < 1.

- Any strong E-term condition (as described in Chapter 4 and introduced in [24, Section 2]) is a height < 1 Mal'tsev condition,
- The Mal'tsev condition of having a semilattice term

$$\mathcal{S} := \{b(x,y) \approx b(y,x), b(x,b(y,z)) \approx b(b(x,y),z), b(x,x) \approx x\}$$

  is *not* of height < 1 (see [41] for a proof).

It follows immediately from the definition that a consistent, nontrivial Mal'tsev condition of height < 1 is equivalent to a condition which contains only equations of the form $x \approx x$ and $f(x_{i_1}, \ldots, x_{i_n}) \approx x_i$, where $i \in \{i_1, \ldots, i_n\}$ and f is a function symbol. The following observations lay out how to obtain a cube term condition from these equations. We first outline the procedure, then state the result (Lemma 5.1.4), and then give an example to illustrate the idea. Before proceeding we remind the reader of the definition of a cube term, as given in Chapter 2.

Definition 5.1.3. Fix $n > 1$, $k > 1$. A term $t(x_1, \ldots, x_n)$ of the algebra $\mathbf{A}$ is a *k-cube term of* $\mathbf{A}$ if it satisfies a system of equations of the form $t(M) \approx \overline{x}$ where M is a $(k \times n)$-matrix over $\{x, y\}$ in which every column contains at least one y, and $\overline{x}$ is the vector $(x, \ldots, x)^T$ all of whose entries are x.

The term $t(x_1, \ldots, x_n)$ is a *cube term of* $\mathbf{A}$ if there is some $k > 1$ such that $t(x_1, \ldots, x_n)$ is a k-cube term of $\mathbf{A}$.

We now outline the steps to demonstrate that every nontrivial strong Mal'tsev condition of height < 1 implies a cube term condition. Let Σ be a strong Mal'tsev condition of height < 1. By collecting together all of the equations involving the function symbol f, we can rewrite Σ as a disjoint collection of matrix conditions of the form $f(M) = \overline{x}$ where M is the matrix whose rows are the patterns of variables occurring in equations $f(x_{i_1}, \ldots, x_{i_n}) \approx x_i$ of Σ and the vector $\overline{x}$ is the vector consisting of the corresponding x_i's. To test satisfaction of Σ it is enough to check satisfaction of each of these matrix conditions separately, since they each involve different function symbols and Σ has no equations relating any two of these symbols. We claim that if the condition $f(M) = \overline{x}$ is consistent and nontrivial, then any algebra satisfying these equations also has a d-cube term of the same arity as f where d is the number of rows of M. Suppose that the variables occurring in the equations $f(M) = \overline{x}$ are $x_1, \ldots, x_K$, where $K \geq 2$.

We begin by permuting the variables of each row of M. Given the equation $f(x_{i_1}, \ldots, x_{i_n}) = x_i$, if $i = i_j \neq 1$, then we permute the variables involved by swapping each occurrence of x_1 with x_i and vice versa.[3] Clearly any operation which satisfies the original equation also satisfies the permuted equation. What we obtain is a condition

[3]Note that x_1 may not appear in the original equation but certainly appears in the permuted equation.

of the form $f(M') = \overline{x}_1$ where M' is a new matrix over $\{x_1, \ldots, x_K\}$ and $\overline{x}_1$ is the vector $(x_1, \ldots, x_1)^T$ all of whose entries are x_1.

If the matrix M' has the vector $\overline{x}_1$ as one of its columns, then by reversing the permutations applied to each row of M, we see that $\overline{x}$ is a column of M. Since we assumed the condition $f(M) = \overline{x}$ was nontrivial, this cannot be the case. Hence the vector $\overline{x}_1$ does not appear as a column of M.

Finally, for each variable x_i appearing in M' with $i > 2$, replace every occurrence of x_i with x_2 to obtain a new condition $f(N) = \overline{x}_1$.

Clearly any operation satisfying the equations $f(M') = \overline{x}_1$ also satisfies these new equations, and the vector $\overline{x}_1$ does not occur as a column of N (so each column of N contains at least one occurrence of x_2). These are the defining conditions of a cube term. Hence any algebra satisfying the condition Σ also has cube terms of the same arities as the function symbols occurring in the nontrivial equations of Σ. We summarize this in the following lemma:

Lemma 5.1.4. *Let Σ be a consistent, nontrivial strong Mal'tsev condition of height < 1 and $\mathbf{A}$ an algebra such that $\mathbf{A}$ satisfies Σ. Then $\mathbf{A}$ has a cube term of arity $ar(f)$ for every function symbol f involved in a nontrivial system of equations of Σ.*

Example 5.1.5. Let Σ be the Mal'tsev condition given by:

$$\Sigma = \{g(w,w,y,y) \approx w, \quad g(x,y,y,z) \approx z, \quad g(x,x,x,w) \approx w, \quad g(w,y,z,y) \approx z, \\ h(x,y,x) \approx y, \ h(x,x,y) \approx y\}.$$

Testing the satisfaction of the Mal'tsev condition Σ can be achieved by checking separately the conditions:

$$g\begin{pmatrix} w & w & y & y \\ x & y & y & z \\ x & x & x & w \\ w & y & z & y \end{pmatrix} \approx \begin{pmatrix} w \\ z \\ w \\ z \end{pmatrix},$$

and

$$h\begin{pmatrix} x & y & x \\ x & x & y \end{pmatrix} \approx \begin{pmatrix} y \\ y \end{pmatrix}.$$

The equations for h already constitute those of a cube term condition (in particular, a 2-cube term of arity 3). The equations for g imply a cube term condition which we can obtain by following the procedure outlined before Lemma 5.1.4. Choosing $x_1 := w$, $x_2 := x$, $x_3 := y$, $x_4 := z$ the equations for g above become:

$$g\begin{pmatrix} w & w & y & y \\ x & y & y & w \\ x & x & x & w \\ z & y & w & y \end{pmatrix} \approx \begin{pmatrix} w \\ w \\ w \\ w \end{pmatrix},$$

after permuting the variables, and then:

$$g\begin{pmatrix} w & w & x & x \\ x & x & x & w \\ x & x & x & w \\ x & x & w & x \end{pmatrix} \approx \begin{pmatrix} w \\ w \\ w \\ w \end{pmatrix}$$

after limiting to only the variables x_1 and x_2. This is clearly a cube term condition (a 4-cube term of arity 4). Note that the choice of $x_1 := w$, $x_2 := x$, $x_3 := y$, $x_4 := z$ was alphabetical only. We could equally well have chosen $x_1 := y$, $x_2 := z$, $x_3 := x$, $x_4 := w$ to arrive at the equivalent condition:

$$g\begin{pmatrix} y & y & z & z \\ z & z & z & y \\ z & z & z & y \\ z & z & y & z \end{pmatrix} \approx \begin{pmatrix} y \\ y \\ y \\ y \end{pmatrix}.$$

5.2 $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma} \in$ NP when Σ is a Strong Mal'tsev Condition of Height < 1

We will now explore how the main result of [13, Subsection 4.4] can be used to solve the problem $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$. Throughout this section we assume that Σ is a strong Mal'tsev condition of height < 1. Without loss of generality[4] we may assume that Σ is the matrix condition $f(M) = \overline{x}$.

Thanks to Lemma 5.1.4 we see that a necessary condition for $\mathbf{A}$ to be a YES instance of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ is that $\mathbf{A}$ has a k-cube term of arity $r :=\mathrm{ar}(f)$ (where M is a $(k \times r)$ - matrix). The next result shows that this condition can be checked in polynomial-time.

Proposition 5.2.1. *Let $\mathbf{A}$ be an algebra and $k > 1$. The following are equivalent:*

1. *$\mathbf{A}$ has a k-cube term of some arity,*
2. *$\mathbf{A}$ has a k-edge term (of arity $(k+1)$),*
3. *$\mathbf{A}$ has a $(1, k-1)$-parallelogram term (of arity $(k+3)$).*[5]

Proof. The equivalence of (1) and (2) is Theorem 2.12 in [8]. Equivalence of the third condition is Theorem 3.5 in [32]. □

Corollary 5.2.2. *Fix $k > 1$. The problem of deciding whether a finite idempotent algebra admits a k-cube term is in P.*

[4] As outlined in the previous section.

[5] See Example 1.2.12.

Proof. By the above theorem, it is enough to check whether $\mathbf{A}$ admits a k-edge term. A polynomial-time algorithm is given in [24, Section 2]. □

By examining the proof of [32, Theorem 3.5], we obtain the following fact which is vital in order to use the result of Bulatov, Mayr, Szendrei:

Proposition 5.2.3. *For any $k \geq 2$ there is a polynomial-time algorithm which, given a finite idempotent algebra $\mathbf{A}$, will return the operation table of a $(1, k-1)$-parallelogram term operation of $\mathbf{A}$ if such a term operation exists (and otherwise return NO).*

Proof. Using Corollary 4.4.7 there is an algorithm to build the operation table of a k-edge term operation of $\mathbf{A}$ which exists if and only if $\mathbf{A}$ has a $(1, k-1)$-parallelogram term. Given the operation table of a k-edge term operation E, we can easily construct the table of the operation $P(x, y, z, w_1, \ldots, w_k)$ defined as

$$E(E(x, w_2, w_2, w_3, \ldots, w_k), E(y, z, w_2, w_2, \ldots, w_2), w_2, \ldots, w_k)$$

which is shown to satisfy the identities defining a $(1, k-1)$-parallelogram term in [32] using the edge identities satisfied by E. □

In [29] it is shown that determining the existence of a minority term for a finite idempotent algebra is an NP problem. We use similar methods together with the result of [13, Subsection 4.4] to prove the following generalisation:

Theorem 5.2.4. *Let Σ be a strong Mal'tsev condition of height < 1. The problem Sat^{Id}_{Σ} is in NP.*

The proof of this theorem involves rephrasing the satisfaction of the Mal'tsev condition Σ as an instance of the subpower membership problem (SMP), and then solving this using Bulatov, Mayr, and Szendrei's result for solving SMP when the algebra involved has a cube term [13, Theorem 4.13]. Here we reproduce the key sections of [13] in order to see that there is only a polynomial dependence on the size of our input to $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$, $\mathbf{A}$.

For the remainder of this section we make the following assumptions: $\mathbf{A}$ is a finite idempotent algebra with a $(1, k-1)$-parallelogram term P of arity $ar(P) = k+3$. The assumption that $\mathbf{A}$ has a parallelogram term can of course be checked in polynomial-time. Indeed, thanks to Proposition 5.2.3 the operation table of such a term operation can be built in polynomial-time. The algorithms outlined in [13] and partially reproduced in this thesis require only terms built from the parallelogram term P, so-called *P-terms*. We introduce the notations $\langle a_1, \ldots, a_n \rangle_P$ to mean the subalgebra of $\mathbf{A}' := (A, P)$ generated by $\{a_1, \ldots, a_n\}$ and $\mathbf{B} \leq_P \mathbf{A}$ to mean that $(B, P^{\mathbf{A}}|_B)$ is a subalgebra of $\mathbf{A}' := (A, P)$. We begin by introducing some of the P-terms used in solving SMP as seen in [13, Subsection 2.2]. Note that the corresponding term operations can also be calculated efficiently given the table for the term operation $P^{\mathbf{A}}$.

Lemma 5.2.5. *[13, Subsection 2.2] Let* $\mathbf{A}$ *be an algebra with a* $(1, k-1)$*-parallelogram term* P*. Then the terms*

$$s(x_1, \dots, x_k) := P(x_1, x_2, x_2, x_1, x_2, x_3, \dots, x_k)$$

and

$$p(x, u, y) := P(x, u, y, x, y, \dots, y)$$

satisfy:

$$y \approx p(x, x, y)$$
$$p(x, y, y) \approx s(x, y, \dots, y)$$
$$s(y, x, y, \dots, y) \approx s(y, y, x, y, \dots, y) \approx \cdots \approx s(y, y, \dots, y, x) \approx y$$

Proof. All of the given equations follow from the $(1, k-1)$-parallelogram identities satisfied by P and the definitions of p and s. □

Following [13] we use x^y to denote the term $p(x, y, y)$. This notation is used in the following definition of *derived forks* which is taken from [13, Section 3].

Definition 5.2.6. Let $\mathbf{B}$ be a subpower of $\mathbf{A}$, say $\mathbf{B} \leq \mathbf{A}^n$. For $i \in [n]$ and $\beta, \gamma \in A$, we define (β, γ) *is a fork in the* i*-th coordinate of* $\mathbf{B}$ if there exist $\overline{b}, \overline{c} \in B$ such that $\overline{b}|_{[i-1]} = \overline{c}|_{[i-1]}$, $\overline{b}|_i = \beta$, and $\overline{c}|_i = \gamma$. We say that $\overline{b}, \overline{c}$ are *witnesses for the fork* (β, γ). The set of all forks in the i-th coordinate of $\mathbf{B}$ is denoted $\mathsf{FORK}_i(B)$. We say that (β, γ) is a *derived fork in the* i*-th coordinate of* $\mathbf{B}$ if there exists $\delta \in A$ such that (β, δ) is a fork in the i-th coordinate of $\mathbf{A}$ and $\gamma = \delta^\beta$ (i.e. the derived forks are forks of the form $(\beta, p(\delta, \beta, \beta))$ where (β, δ) is a fork). The set of all derived forks in the i-th coordinate of $\mathbf{B}$ is denoted $\mathsf{FORK}'_i(B)$. These definitions will also be used for sub*sets* of powers of A (not just subpowers of $\mathbf{A}$).

The following is Lemma 3.2 in [13], rephrased a little in our restricted setting. The lemma shows that the derived forks of $\mathbf{B}$ are also forks of $\mathbf{B}$ with an extra special property. We will see in later algorithms that having "designated witnesses" for these special forks will be helpful in solving $\mathrm{SMP}(\mathbf{A})$ (and hence $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$). For a proof of the following lemma, we refer the reader back to [13].

Lemma 5.2.7. *Let* $\mathbf{B}$ *be a subpower of* $\mathbf{A}$*, say* $\mathbf{B} \leq \mathbf{A}^n$*. Then:*

1. *For every* $(\gamma, \delta) \in \mathsf{FORK}'_i(B)$ *and for every* $\overline{b} \in B$ *with* $\overline{b}|_i = \gamma$*, there is an element* $\overline{b}' \in B$ *such that* $\overline{b}, \overline{b}'$ *witness that* $(\gamma, \delta) \in \mathsf{FORK}_i(B)$
2. *Hence* $\mathsf{FORK}'_i(B) \subseteq \mathsf{FORK}_i(B)$*.*

Before proceeding, we give a small example of how forks can be used. The example is drawn from [11, Lemma 3.1] and the remarks preceding that result, and is reliant upon the presence of a Mal'tsev term for the algebra of interest. With only a parallelogram term to work with the constructions become a little more intricate but the Mal'tsev example provides a convenient illustration of the relevant observation.

Example 5.2.8. Suppose $\mathbf{A}$ is a finite algebra with a Mal'tsev term $t(x, y, z)$. Let $\overline{a}, \overline{b}, \overline{c} \in B \subseteq A^n$ such that $\overline{a}, \overline{b}$ are witnesses for the fork $(\overline{a}|_n, \overline{b}|_n) \in \mathsf{FORK}_n(B)$ and $\overline{c}|_n = \overline{b}|_n$.

It follows that the tuple $\overline{d}$ which satisfies $\overline{d}|_{[n-1]} = \overline{c}|_{[n-1]}$ and $\overline{d}|_n = \overline{a}|_n$ is in the subalgebra generated by $\{\overline{a}, \overline{b}, \overline{c}\}$ because:

$$t(\overline{c}, \overline{b}, \overline{a})|_{[n-1]} = t(\overline{c}|_{[n-1]}, \overline{b}|_{[n-1]}, \overline{a}|_{[n-1]}) = t(\overline{c}|_{[n-1]}, \overline{b}|_{[n-1]}, \overline{b}|_{[n-1]}) = \overline{c}|_{[n-1]}$$

and

$$t(\overline{c}, \overline{b}, \overline{a})|_n = t(\overline{c}|_n, \overline{b}|_n, \overline{a}|_n) = t(\overline{c}|_n, \overline{c}|_n, \overline{a}|_n) = \overline{a}|_n.$$

The reason for introducing the technical definitions in this subsection is to use observations similar to those of the last example (Example 5.2.8) to find alternative generating sets for particular subpowers of $\mathbf{A}$, from which every element of the generated subpower can be calculated efficiently. In [11], Bulatov and Dalmau use such generating sets to provide an algorithm for solving CSPs over a Mal'tsev template (i.e. when the constraint relations are all invariant under some Mal'tsev operation). The authors of [8] and [25] extend this result to include templates invariant under any cube term condition (equivalently- to include any algebra which generates a variety with few subpowers[6]). In [36] Mayr uses the technique to show that SMP($\mathbf{A}$) is in NP whenever $\mathbf{A}$ is a finite algebra with a Mal'tsev term, and in [13], together with Bulatov and Szendrei, they extend this result to include all finite algebras with cube terms. The authors of [29] used Mayr's original result for Mal'tsev algebras to derive an NP algorithm for $\mathrm{Sat}^{\mathrm{Id}}_{\mathrm{Minority}}$ and to complete the pattern we now extend that result to the case of arbitrary cube terms. Known in the literature as "representations" ([8], [11], [13], [25], [36], [29]), we save the definition of these generating sets until after these lemmata, which motivate the particular choice of definition in this instance. The first of these demonstrates that any element $\overline{b}$ of a given subpower $\mathbf{B} \leq \mathbf{A}^n$ can be obtained using a particular P-term applied to certain tuples which agree with $\overline{b}$ on particular coordinates.

Lemma 5.2.9. *For every $n \geq k$ there exists a P-term $t_n = t_n(x, y, z, \overline{w})$ where $\overline{w}$ is a tuple of variables indexed by $\binom{[n]}{k-1}$ (hence of length $\binom{n}{k-1}$) such that:*

1. *For every subset $R \subseteq A^n$ and for every tuple $\overline{b} = (b_1, \ldots, b_n)$ in A^n, if:*
   - *(i) For each $J \in \binom{[n]}{k-1}$ the set R contains a tuple $\overline{b^J}$ satisfying $\overline{b^J}|_J = \overline{b}|_J$, and*
   - *(ii) For some element $\overline{b}' = (b_1, \ldots, b_{n-1}, \beta)$ of the P-subalgebra*

$$\mathbf{R}^* := \langle R \rangle_P \leq_P \mathbf{A}^n,$$

[6] See [25] for a definition of this property which is shown in that paper to be equivalent to the Mal'tsev condition of having a k-edge term for some k (and equivalently to satisfying some cube term condition).

the set R contains tuples

$$\overline{u} = (u_1, \ldots, u_{n-1}, b_n)$$

and

$$\overline{u}' = (u_1, \ldots, u_{n-1}, \beta^{b_n})$$

which are witnesses for the fork $(b_n, \beta^{b_n}) \in \mathsf{FORK}_n(R)$,

then

$$\overline{b} = t_n(\overline{b}', \overline{u}', \overline{u}, (\overline{b^J})_{J \in \binom{[n]}{k-1}}) \tag{5.1}$$

and therefore $\overline{b} \in \mathbf{R}^*$

2. *There is a P-circuit C for t_n which is of size $O(n^k)$, and there is a polynomial-time algorithm which takes as input n and outputs C in time $O(n^k)$.*[7]

Proof. We reproduce the proof seen in [13, Lemma 3.8] to verify that the construction of a P-circuit for the term t_n is independent of the algebra $\mathbf{A}$. The proof works by inductively building a term which satisfies Equation 5.1 when projected onto certain subsets of the index set $[n]$.

We show by induction that for each $l = n, n-1, \ldots, k-1$ and every $V' \in \binom{[l]}{k-1}$ there is a P-term $t_{l,V'}(x, y, z, (w^J)_{J \in \binom{V}{k-1}})$, where $V = V' \cup \{l+1, \ldots, n\}$ such that whenever $R \subseteq A^n$, $\overline{b} \in A^n$, $(\overline{b^J})_{J \in \binom{[n]}{k-1}}$, $\overline{b}'$ $\overline{u}$, $\overline{u}'$ satisfy the conditions of the lemma, then $t_{l,V'}$ satisfies:

$$\overline{b}|_V = t_{l,V'}(\overline{b}', \overline{u}', \overline{u}, (\overline{b^J})_{J \in \binom{V}{k-1}})|_V$$

For the base case when $l = n$ we have $V = V' \in \binom{[n]}{k-1}$ and choosing

$$t_{l,V'}(x, y, z, (w_J)_{J \in \binom{V}{k-1}}) = w_{V'}$$

we see that the equation is true by the assumption that $\overline{b^{V'}}|_{V'} = \overline{b}|_{V'}$ given in the statement of the lemma.

For the inductive step, suppose that the statement holds for $l+1$ where

$$k - 1 \leq l < n.$$

We show that the statement is true for l. Suppose

$$V' = \{i_1, \ldots, i_{k-1}\} \in \binom{[l]}{k-1},$$

[7] In order to verify Equality 5.1 it would be necessary to run the constructed P-circuit on n-tuples of entries which would therefore have a time requirement of $O(n^{k+1})$.

$$V := V' \cup \{l+1, \ldots, n\},$$

and for each $j = 1, \ldots, k-1$,

$$V'_j := \{i_1, \ldots, i_{j-1}, i_{j+1}, \ldots, i_{k-1}, l+1\} \in \binom{[l+1]}{k-1},$$

and

$$V_j := V \backslash \{i_j\} = V'_j \cup \{l+2, \ldots, n\}.$$

Define $t_{l,V'}(x, y, z, (w^J)_{J \in \binom{V}{k-1}}) :=$

$$P\Big(\; s\big(\; s\big(x, (t_{l+1,V'_j}(x, y, z, (w^J)_{J \in \binom{V_j}{k-1}}))_{j \in [k-1]} \; \big) \; , \; (t_{l+1,V'_j}(x, y, z, (w^J)_{J \in \binom{V_j}{k-1}}))_{j \in [k-1]} \; \big), \\ p\big(y, z, t_{l+1,V'_1}(x, y, z, (w^J)_{J \in \binom{V_1}{k-1}})\big), \; t_{l+1,V'_1}(x, y, z, (w^J)_{J \in \binom{V_1}{k-1}}), \; x \; , \\ (t_{l+1,V'_j}(x, y, z, (w^J)_{J \in \binom{V_j}{k-1}}))_{j \in [k-1]} \; \Big)$$

It is shown in [13] that given $R \subseteq A^n$, $\bar{b} \in A^n$, $(\overline{b^J})_{J \in \binom{[n]}{k-1}}$, $\overline{b}'$ $\overline{u}$, $\overline{u}'$ satisfying the conditions of the lemma, the term $t_{l,V'}$ defined above does indeed satisfy $\bar{b}|_V = t_{l,V'}(\overline{b}', \overline{u}', \overline{u}, (\overline{b^J})_{J \in \binom{V}{k-1}})|_V$. That part of the proof is not reproduced here but can be verified by the more diligent reader using only the induction hypothesis and the identities satisfied by P, s and p outlined in Example 1.2.12 and Lemma 5.2.5. It follows by induction that the term $t_n := t_{k-1,[k-1]}$ satisfies Equation 5.1 and this proves the first part of the lemma.

Note that we cannot necessarily construct an operation table for $t_{l,V'}$ in polynomial-time (in $||\mathbf{A}||$) from the given tables for t_{l+1,V'_j} because the arity of the operations depends on n which will later be fixed as $|A|^k$. To store an operation table of arity $K := 3 + \binom{|A|^k}{k-1}$ there is a space requirement of $|A|^K$ which is $\Omega(|A|^{|A|^{k(k-1)}})$.[8]

Given any particular input, a, b, c, $(d^J)_{J \in \binom{[n]}{k-1}}$, to t_n however, the inductive argument above allows us to calculate $t_n(a, b, c, (d^J)_{J \in \binom{[n]}{k-1}})$ in polynomial-time. As observed in [13], the term t_n has $\sum_{l=k-1}^{n-1} \binom{l}{k-1} = O(n^k)$ distinct subterms of the form $t_{l,V'}$ where $l = k-1, \ldots, n$ and $V' \in \binom{[l]}{k-1}$. Given the value of the subterms $t_{l+1,W'}$ on the input a, b, c, $(d^J)_{J \in \binom{W}{k-1}}$ for each $W \in \binom{[l+1]}{k-1}$, one can determine the value of each $t_{l,V'}$ using four applications of the term P (shown in Equation 5.2 as P, s (twice), and p). Since an operation table for P is given, each of these table lookups is achieved in constant time. Hence the inductive procedure outlined above describes how to construct and implement a P-circuit for t_n in time $O(n^k)$. □

[8] We say that $f(n)$ is $\Omega(g(n))$ if there exist positive integers c and N such that for all $n > N$ we have $f(n) \geq cg(n)$.

The following example illustrates the implementation of such a circuit in the group S_5.

Example 5.2.10. Recall that in a group $\mathbf{G} := \langle G, \cdot, ^{-1}, 1\rangle$ the term $p(x,y,z) := xy^{-1}z$ is a Mal'tsev (and hence 2-edge) term. It follows that the term $P(x,y,z,u,v) := xy^{-1}z$ is a (1,1)-parallelogram term in any group. If we were to explicitly construct the term derived in the above lemma even for $n = 4$ using this parallelogram term (so with $k = 2$), we arrive at the rather unwieldy expression:

$$t_4(x,y,z,w^{\{1\}},w^{\{2\}},w^{\{3\}},w^{\{4\}}) :=$$
$$P(x,p(y,z,P(x,p(y,z,P(x,p(y,z,w^{\{4\}}),w^{\{4\}},x,w^{\{4\}})),$$
$$P(x,p(y,z,w^{\{4\}}),w^{\{4\}},x,w^{\{4\}}),x,P(x,p(y,z,w^{\{4\}}),w^{\{4\}},x,w^{\{4\}}))),$$
$$P(x,p(y,z,P(x,p(y,z,w^{\{4\}}),w^{\{4\}},x,w^{\{4\}})),P(x,p(y,z,w^{\{4\}}),w^{\{4\}},x,w^{\{4\}}),x,$$
$$P(x,p(y,z,w^{\{4\}}),w^{\{4\}},x,w^{\{4\}})),x,P(x,p(y,z,P(x,p(y,z,w^{\{4\}}),w^{\{4\}},x,w^{\{4\}})),$$
$$P(x,p(y,z,w^{\{4\}}),w^{\{4\}},x,w^{\{4\}}),x,P(x,p(y,z,w^{\{4\}}),w^{\{4\}},x,w^{\{4\}}))),$$

where the colours have been chosen to represent the subterms built at each level of the inductive process outlined. Of course, to calculate the value of the term on any given input we would not need to know this direct expression. Suppose for example we are working in the group S_5 of permutations of the set $\{1,2,3,4,5\}$ and we are given the tuples:

$$\overline{b}^{\{1\}} := \begin{pmatrix}(12)\\(12)\\(12)\\(12)\end{pmatrix}, \overline{b}^{\{2\}} := \begin{pmatrix}(123)\\(123)\\(123)\\(123)\end{pmatrix}, \overline{b}^{\{3\}} := \begin{pmatrix}(1234)\\(1234)\\(1234)\\(1234)\end{pmatrix}, \overline{b}^{\{4\}} := \begin{pmatrix}(12345)\\(12345)\\(12345)\\(12345)\end{pmatrix}, \overline{b}' := \begin{pmatrix}(12)\\(123)\\(1234)\\(123)(45)\end{pmatrix}, \overline{u}' := \begin{pmatrix}(123)\\(123)\\(123)\\(123)(45)\end{pmatrix}, \text{ and } \overline{u} := \begin{pmatrix}(123)\\(123)\\(123)\\(12345)\end{pmatrix}$$

which satisfy the conditions of the lemma for $n = 4$ and $\overline{b} := \begin{pmatrix}(12)\\(123)\\(1234)\\(12345)\end{pmatrix}$.

We know that if we input these tuples to the term t_4, then the output will be $\overline{b}$. Without knowing this we could still calculate the value of t_4 on these tuples efficiently using the circuit described by the inductive argument in the proof of the lemma. The calculation runs as follows:

- $t_{4,\{i\}}(\overline{b}', \overline{u}', \overline{u}, \overline{b}^{\{i\}}) = \overline{b}^{\{i\}}$ where $i = 1, \ldots, 4$ (this exhausts all the $(k-1)$-element subsets of $[4]$)

- $t_{3,\{i\}}(\overline{b}', \overline{u}', \overline{u}, \overline{b}^{\{i\}}, \overline{b}^{\{4\}}) = P(\overline{b}', p(\overline{u}', \overline{u}, \overline{b}^{\{4\}}), \overline{b}^{\{4\}}, \overline{b}', \overline{b}^{\{4\}}) = \begin{pmatrix}(12)\\(123)\\(1234)\\(12345)\end{pmatrix}$ where $i = 1, 2, 3$
- $t_{2,\{i\}}(\overline{b}', \overline{u}', \overline{u}, \overline{b}^{\{i\}}, \overline{b}^{\{3\}}, \overline{b}^{\{4\}})$
  $= P(\overline{b}', p(\overline{u}', \overline{u}, \begin{pmatrix}(12)\\(123)\\(1234)\\(12345)\end{pmatrix}), \begin{pmatrix}(12)\\(123)\\(1234)\\(12345)\end{pmatrix}, \overline{b}', \begin{pmatrix}(12)\\(123)\\(1234)\\(12345)\end{pmatrix}) = \begin{pmatrix}(12)\\(123)\\(1234)\\(12345)\end{pmatrix}$ where $i = 1, 2$
- $t_{1,\{1\}}(\overline{b}', \overline{u}', \overline{u}, \overline{b}^{\{1\}}, \overline{b}^{\{2\}}, \overline{b}^{\{3\}}, \overline{b}^{\{4\}})$
  $= P(\overline{b}', p(\overline{u}', \overline{u}, \begin{pmatrix}(12)\\(123)\\(1234)\\(12345)\end{pmatrix}), \begin{pmatrix}(12)\\(123)\\(1234)\\(12345)\end{pmatrix}, \overline{b}', \begin{pmatrix}(12)\\(123)\\(1234)\\(12345)\end{pmatrix}) = \overline{b}$ as required.

Each step of this calculation requires only table lookups in the table for P (plus the initial projection operations $t_{4,\{i\}}$). The values of the subterm calculations are used in subsequent calculations and appear with the same colours used in the syntactic description of the term given above. In total we only needed to reference $4 \cdot (2 \cdot 3 + 2 \cdot 2 + 2 \cdot 1) = 48$ values of the term operation P. In this simple example we actually performed many more calculations than was necessary. Since $k = 2$ the terms $t_{l,\{i\}}$ are always independent of the subset $\{i\}$ (because for any choice of $V' = \{i\}$ we have $V'_1 = \{l+1\}$). For larger values of k the terms $t_{l,V'}$ clearly do depend on the choice of V'. The calculation was also simplified by the observation that when $k = 2$ the term $s(x_1, x_2) := P(x_1, x_2, x_2, x_1, x_2)$ satisfies $s(x_1, x_2) \approx x_1$.

A graphical representation of the circuit constructed in the proof of Lemma 5.2.9 for the term $t_4(x, y, z, w^{\{1\}}, w^{\{2\}}, w^{\{3\}}, w^{\{4\}})$ is given in Figure 5.1. Note that each p-gate is really just a P-gate in which two of the inputs are duplicated. Notice also that the constructed circuit works for any algebra with a $(1,1)$-parallelogram term and is not specific to the algebra S_5. Again we remind the reader that circuits for $k > 2$ are more complicated since in this case the term s is not necessarily a projection and the terms $t_{l,V'}$ generally do depend on the subset V'.

A practical application of the above lemma clearly relies on knowing the tuple $\overline{b}' \in \mathbf{R}^*$ which agrees with $\overline{b}$ in all coordinates except the last. However if $n = k$, then the element $\overline{b}^{[k-1]}$ can be taken as $\overline{b}'$ and we need only the "fork witnesses" $\overline{u}$ and $\overline{u}'$. This suggests that for $n > k$ the tuple $\overline{b}$ could be obtained through recursively creating better approximations. The next lemma will show that given elements witnessing the right forks, we can construct $\overline{b}$ as a single term applied to these witnesses, again in an

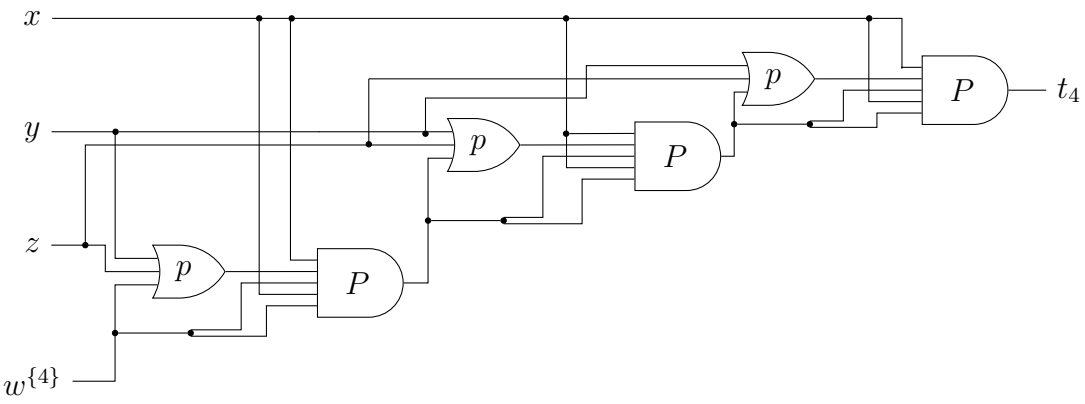

Figure 5.1: A Circuit for the Term $t_4 := t_4(x, y, z, w^{\{1\}}, w^{\{2\}}, w^{\{3\}}, w^{\{4\}})$

efficient manner. Before stating the lemma, we introduce the P-terms involved. For each $m = k-1, k, \ldots, n$ we define T_m as follows:

- $T_{k-1}(w_{[k-1]}) := w_{[k-1]}$
- For $k \leq m \leq n$

$$T_m(z^{(k)}, \hat{z}^{(k)}, \ldots, z^{(m)}, \hat{z}^{(m)}, \overline{w}^{[m]}) :=$$
$$t_m(T_{m-1}(z^{(k)}, \hat{z}^{(k)}, \ldots, z^{(m-1)}, \hat{z}^{(m-1)}, \overline{w}^{[m-1]}), z^{(m)}, \hat{z}^{(m)}, \overline{w}^{[m]})$$

  where $\overline{w}^S := (w_J)_{J \in \binom{S}{k-1}}$ is a tuple of variables indexed by the $k-1$ element subsets of the set S, and t_m is the term introduced in Lemma 5.2.9.

Once again we see that the arity of the terms T_m increases as we increase the value of m. Nevertheless we may still effectively calculate outputs of T_n using an efficient circuit. Given an input $a^{(k)}, \hat{a}^{(k)}, \ldots, a^{(n)}, \hat{a}^{(n)}, \overline{b}^{[n]}$ to T_n, we need to calculate $T_m(a^{(k)}, \hat{a}^{(k)}, \ldots, a^{(m)}, \hat{a}^{(m)}, \overline{b}^{[m]})$ for $n-k$ different values of m. Using previously stored values of t_j for $j < m$, each of these computations can be performed in time $O(n^k)$ for a total time to compute $T_n(a^{(k)}, \hat{a}^{(k)}, \ldots, a^{(n)}, \hat{a}^{(n)}, \overline{b}^{[n]})$ in the order of $O(n^{k+1})$. The value of obtaining circuits for these terms is seen in [13, Lemma 3.10] and rephrased for our purposes in the next lemma.

Lemma 5.2.11. *Let $n \geq k$, $\overline{b} = (b_1, \ldots, b_n) \in A^n$, and $R \subseteq A^n$ such that:*

1. *For each $J \in \binom{[n]}{k-1}$ the set R contains an element $\overline{b^J}$ satisfying $\overline{b^J}|_J = \overline{b}|_J$, and*
2. *for every $k \leq m \leq n$ the set R contains elements $\overline{u}^{(m)}, \overline{v}^{(m)}$ witnessing the fork $(b_m, \beta^{b_m}) \in \mathsf{FORK}_m(R)$ where*

$$\beta := T_{m-1}(\overline{u}^{(k)}, \overline{v}^{(k)}, \ldots, \overline{u}^{(m-1)}, \overline{v}^{(m-1)}, (\overline{b^J})_{J \in \binom{[m-1]}{k-1}})|_m. \tag{5.2}$$

Then

$$\overline{b}|_{[m]} = T_m(\overline{u}^{(k)}, \overline{v}^{(k)}, \ldots, \overline{u}^{(m)}, \overline{v}^{(m)}, (\overline{b^J})_{J \in \binom{[m]}{k-1}})|_{[m]} \tag{5.3}$$

holds for each $k-1 \leq m \leq n$*, and hence*

$$\overline{b} = T_n(\overline{u}^{(k)}, \overline{v}^{(k)}, \ldots, \overline{u}^{(n)}, \overline{v}^{(n)}, (\overline{b^J})_{J \in \binom{[n]}{k-1}}) \tag{5.4}$$

Proof. The result follows by induction using the definition of T_m and Lemma 5.2.9. For details, see [13, Lemma 3.10]. □

With these terms in hand, we are now ready to define the special generating sets or "representations" mentioned earlier in this section.

Definition 5.2.12. (See [13, Definitions 3.4, 4.2, 4.4, 4.5]) Let $\mathbf{B} \leq \mathbf{A}^n$ be a subpower of $\mathbf{A}$. A *partial standardised representation* (PSR) of $\mathbf{B}$ is given by a subset $R \subseteq B \subseteq A^n$ together with a (surjective) *designation function* $\mathsf{des} : D \twoheadrightarrow R$, where $D \subseteq \{(J, \overline{b}) \mid J \in \binom{[n]}{k-1}, \overline{b} \in A^J\} \sqcup \bigcup_{i=k}^{n} (\{i\} \times \mathsf{FORK}_i(R) \times [2])$ satisfying:

1. Whenever $\mathsf{des}((J, \overline{b})) = \overline{r} \in R$ we must also have $\overline{r}|_J = \overline{b}$ and in this case we say $\overline{r}$ is the *designated local witness for* $\overline{b} \in \mathbf{B}|_J$.

2. If $\mathsf{des}((i, (\alpha, \beta), 1)) = \overline{r}$ and $\mathsf{des}((i, (\alpha, \beta), 2)) = \overline{s}$, then we must also have $\overline{r}|_{[i-1]} = \overline{s}|_{[i-1]}$ and $\overline{r}|_i = \alpha$, $\overline{s}|_i = \beta$. In this case we say that the pair $(\overline{r}, \overline{s})$ are *designated fork witnesses for* $(\alpha, \beta) \in \mathsf{FORK}_i(B)$.

The requirement that des be onto guarantees that every element of R has at least one designation. We typically suppress the designation function and talk of the PSR R (understanding that every element has a designation and this can be formulated as a function in the manner described above). We say that R *has a full set of designated local witnesses for* $\mathbf{B}$ if $\{(J, \overline{b}) \mid J \in \binom{[n]}{k-1}, \overline{b} \in B|_J\}$ is a subset of the domain D (and hence every element of B has a designated witness for every projection onto $k-1$ indices). As in [13] we define the *size of the PSR* R, $||R||$,[9] to be the size of the domain D, and note that this is bounded above by:

$$\binom{n}{k-1}(|A|^{k-1}) + (n-k)(2|A|^2) \tag{5.5}$$

which is $O(n^{k-1}||\mathbf{A}||^{k-1})$.

We say that $\overline{b} = (b_1, \ldots, b_n) \in B$ is *representable by* R if:

1. for every choice of indices $J \in \binom{[n]}{k-1}$, R contains an element $\overline{b^J}$ which is a designated local witness for $\overline{b}|_J \in \mathbf{B}|_J$.

[9] We use the notation $||R||$ to be clear that we are talking about the size of R as a PSR and not just the cardinality of the set R.

2. for every $k \leq m \leq n$ R contains elements $\overline{u}^{(m)}, \overline{v}^{(m)}$ designated to witness the derived fork (b_m, β^{b_m}) where $\beta = T_{m-1}(\overline{u}^{(k)}, \overline{v}^{(k)}, \ldots, \overline{u}^{(m-1)}, \overline{v}^{(m-1)}, (\overline{b^J})_{J \in \binom{[m-1]}{k-1}})|_m$.

$\overline{b}$ is *completely representable by* R if, in addition, there are also designated witnesses in R for every (non-derived) fork (b_m, β) where β is defined as above.

We now have all of the necessary ingredients to prove $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma} \in \mathsf{NP}$. Recall that throughout this section we have been assuming that Σ is a consistent strong Mal'tsev condition of height < 1. The following proof of Theorem 5.2.4 demonstrates how to find a polynomial-time certificate for YES instances of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$.

Proof of Theorem 5.2.4. Let $\mathbf{A}$ be an instance of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$. We will construct an equivalent instance $\overline{a}_1, \ldots, \overline{a}_n, \overline{b}$ of SMP($\mathbf{A}$) which we can solve using the result of Bulatov, Mayr, Szendrei [13, Theorem 4.13]. We outline the algorithm in some detail to be sure that the dependence on the size of the algebra $||\mathbf{A}||$ is polynomial.

As noted in the discussion of height < 1 Mal'tsev conditions in the previous section, we may assume without loss of generality that Σ is a single matrix condition $f(M) \approx \overline{x}$ involving only one function symbol f of arity $ar(f) = r$. Let k be the number of rows of M and recall that a necessary condition for $\mathbf{A}$ to satisfy Σ is that $\mathbf{A}$ has a $(1, k-1)$-parallelogram term thanks to Lemma 5.1.4 and Proposition 5.2.1.

Our first step then is to run the algorithms outlined in Corollary 4.4.7 and Proposition 5.2.3 to build the operation table of a $(1, k-1)$-parallelogram term operation P of $\mathbf{A}$. If no such term operation exists, then we return the answer NO and halt the algorithm- $\mathbf{A}$ does not satisfy Σ. Using Corollary 4.4.7 we can build an edge term operation in time $O(||\mathbf{A}||^{4k+3})$ and then Proposition 5.2.3 gives us the table of a $(1, k-1)$-parallelogram term operation in time $O(|A|^{k+3})$ which is $O(||\mathbf{A}||^{k+3})$. Hence this first step requires time $O(||\mathbf{A}||^{4k+3})$.

From now on, we assume that $\mathbf{A}$ has a $(1, k-1)$-parallelogram term and P is an operation table for such a term operation of $\mathbf{A}$.

Our next step is to build an appropriate instance of SMP($\mathbf{A}$) and then describe the algorithms used in a polynomial verifier for the constructed instance. We need the following definition which appeared in Chapter 3 (Definition 3.1.6) and is briefly recalled here. We say that $\overline{a} \in A^r$ *matches the equality pattern of* $\overline{x} \in \{x_1, \ldots, x_K\}^r$ if for each $1 \leq i < j \leq r$ we have $\overline{x}|_i = \overline{x}|_j \implies \overline{a}|_i = \overline{a}|_j$.

An operation $g : A^r \to A$ satisfies the Mal'tsev condition Σ if and only if whenever $\overline{a} = (a_1, \ldots, a_r) \in A^r$ matches the equality pattern of row M_j of M, we have $g(a_1, \ldots, a_r) = a_{i_j}$, where $\overline{x}|_j = x_{i_j}$.

Let I be the subset of A^r defined by

$$\{\overline{a} \in A^r \mid \text{for some } j = 1, \ldots, k, \ \overline{a} \text{ matches the equality pattern of the row } M_j \text{ of } M\}.$$

Note that there is only one function $\hat{g} : I \to A$ which "satisfies" Σ, since whenever $\overline{a} \in A^r$ matches the equality pattern of some row of M, the output of the function $\hat{g}$

on that tuple is determined by the equations $f(M) \approx \overline{x}$. If $\overline{a}$ matches the equality pattern of more than one row of M (which is certainly possible), then the consistency of Σ guarantees that the prescribed outputs from each row are also equal. Recall that r-ary partial operations on A with domain $D \subseteq A^r$ can be represented by tuples in A^D and let $g \in A^I$ be the tuple representing the unique function $\hat{g} : I \to A$ which "satisfies" Σ.

Now consider the functions $\hat{\pi}_i : I \to A : (a_1, \ldots, a_r) \mapsto a_i$ (restricted projection functions). Let $\pi_i \in A^I$ $(i = 1, \ldots, r)$ be the corresponding tuples. From fundamental properties of terms and subalgebra generation (see for example [15, Definition 10.4]), it follows that $\mathbf{A}$ has a term satisfying Σ if and only if $g \in \langle \pi_1, \ldots, \pi_r \rangle_{A^I}$. Hence we have proven the following:

Claim: $\mathbf{A}$ is a YES instance of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma} \iff g, \pi_1, \ldots, \pi_r$ is a YES instance of SMP($\mathbf{A}$).

Each of the tuples $g, \pi_1, \ldots, \pi_r$ are of length $|I| \leq |A|^r$ and the time taken to build them is clearly dominated by the time taken to build I (since every time we include a new element in I we can simultaneously extend each of the tuples to reflect where this element is mapped to under the corresponding functions). To construct the set I we need to check for each tuple $\overline{a} \in A^r$ whether it matches the equality pattern of any row of M. Given a tuple $\overline{a}$ this procedure is clearly linear in r and independent of $||\mathbf{A}||$. Hence the time required to build all of the tuples is $O(|A|^r)$ which is $O(||\mathbf{A}||^r)$.

Our final step is to demonstrate that the algorithms outlined in [13] to prove their Theorem 4.13, can be used to build a polynomial-time verifier for the constructed instance with respect to $||\mathbf{A}||$ (and not just with respect to the size of the input $g, \pi_1, \ldots, \pi_r$).

The proof proceeds as follows (following [13, Theorem 4.13]):

1. Build a partial standardized representation (PSR), R, for $\mathbf{B} := \langle \pi_1, \ldots, \pi_r \rangle_{A^I}$ with a full set of designated local witnesses;

2. Extend R so that each of the generators π_i is completely representable with respect to the extension R';

3. Use an NP oracle to both determine whether there are elements of B which are not completely representable by the constructed representation R' and simultaneously extend R' to correct these deficiencies;

4. Verify that $g, \pi_1, \ldots, \pi_r$ is a YES instance of SMP($\mathbf{A}$) by checking that g is completely representable by the final extension of R'.

We need to show that each of these steps can be achieved with only polynomial-time dependence on $||\mathbf{A}||$. All of the techniques used are those of [13]. The first item requires no technical machinery, whereas the other steps are fairly involved. Item 1 is achieved by the following simple algorithm, adapted from [13, Algorithm 2]:

Algorithm 1: Generating a PSR

Input: A finite algebra $\mathbf{C}$ and $\overline{c}_1, \ldots, \overline{c}_s$, tuples in C^n for some n
Output: A partial standardized representation of $\mathbf{B} := \langle \overline{c}_1, \ldots, \overline{c}_s \rangle_{\mathbf{C}^n}$ which contains a full set of designated local witnesses.
1: $R := \emptyset$
2: **for** each $J \in \binom{[n]}{k-1}$ **do**
3: generate $\mathbf{B}|_J := \langle \overline{c}_1|_J, \ldots, \overline{c}_s|_J \rangle_{A^J}$ and simultaneously for each new element $h(\overline{c}_1|_J, \ldots, \overline{c}_s|_J)$, add $h(\overline{c}_1, \ldots, \overline{c}_s) \in \mathbf{B}$ to R as a designated witness to $h(\overline{c}_1|_J, \ldots, \overline{c}_s|_J) \in \mathbf{B}|_J$.
4: **endfor**
5: **output:** R

We first analyze the complexity of Algorithm 1 in terms of $||\mathbf{C}||$ and $n \in \mathbb{N}$ before specifying the particular input on which we will run the algorithm. Correctness of the algorithm is immediate from the definition of partial standardized representation (Definition 5.2.12).

To calculate the complexity of the algorithm, note that the **for** loop in Step 2 is executed $\binom{n}{k-1}$ times, which is $O(n^{k-1})$.

The result of [20, Proposition 6.1] tells us that the subalgebra $\mathbf{B}|_J$ can be generated in time $O((||\mathbf{C}||)||\mathbf{C}^J||) = O(||\mathbf{C}||^k)$ since J is an index set of size $k-1$. That result is based on closing the set $\{\overline{c}_1|_J, \ldots, \overline{c}_s|_J\}$ under all of the basic operations of $\mathbf{C}$ using any appropriate method which will not calculate the same basic operation applied to the same tuples more than once. In this way, every basic operation f_i of arity r_i is applied precisely once to every tuple in $(B|_J)^{r_i}$. We modify this procedure slightly in the next paragraph to achieve the second half of Step 3 as well.

Rather than applying the basic operations of $\mathbf{C}$ to the tuples $\overline{c}_1|_J, \ldots, \overline{c}_s|_J$ and the subsequently generated J-tuples, if we instead apply the basic operations to the tuples $\overline{c}_1, \ldots, \overline{c}_s$ being sure not to apply the same basic operation to any tuples which share the same projection onto the set C^J more than once, then this will of course increase the runtime. Applying an l-ary basic operation h of $\mathbf{C}$ to every l-tuple over some set $S \subseteq C$ can be done in time $O(l|S|^l)$ (since there are $|S|^l$-many l-tuples and we require $O(l)$ time to read each one with some constant time to look up the value of h on that l-tuple). If instead we were looking at every l-tuple over a subset $T \subseteq C^J$ then there are again $|T|^l \leq (|C^J|^l)$ many l-tuples in the set and we now require $O(l|J|)$ time to read each tuple and $O(|J|)$ time to look up each value.

What we actually need to do is to apply h to all of those l-tuples over C^n which have different projections onto C^J. There are still only as many as $(|C^J|)^l$ such l-tuples to consider but we now require $O(ln)$ time to read each one and $O(n)$ time to look up each value. This process is repeated for every basic operation of $\mathbf{C}$ until the full subalgebra $\mathbf{B}|_J$ has been generated in the appropriate coordinates. Since we were careful not to apply the same basic operation to tuples sharing the same projection onto C^J, the time required to run Step 3 of the algorithm is $O(||\mathbf{C}^J||(Ln))$ (where L is the maximum

arity of the basic operations of $\mathbf{C}$)[10] which is $O(||\mathbf{C}||^{k-1}||\mathbf{C}||n) = O(n||\mathbf{C}||^k)$.

Hence the runtime of Algorithm 1 is $O(n^{k-1}(n||\mathbf{C}||^k)) = O(n^k||\mathbf{C}||^k)$.

We will run this algorithm on the input given by our instance of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$, $\mathbf{A}$, together with the tuples $\pi_1, \dots, \pi_r$ encoding the restricted projection operations, hence with $n := |I|$.[11] In this case, and using the estimate $|I| \leq |A|^r$, we calculate the runtime to be $O((|A|^r)^k||\mathbf{A}||^k)$ which is $O(||\mathbf{A}||^{k(1+r)})$, a polynomial in $||\mathbf{A}||$.

Next we need to extend the constructed PSR R in order to guarantee that each of the generators π_i is completely representable by the extended PSR. This is achieved by Algorithm 2 which is adapted from [13, Algorithm 1].

Algorithm 2: Obtaining Representable Generators

Input: A finite algebra $\mathbf{C}$ which admits a k-cube term, the operation table $P^{\mathbf{C}}$ of a parallelogram term operation of $\mathbf{C}$, tuples $\overline{c}_1, \dots, \overline{c}_s$ in C^n, and a PSR R of $B := \{\overline{c}_1, \dots, \overline{c}_s\}$ with a full set of designated local witnesses for B.
Output: A new PSR $R' \supseteq R$ with respect to which $\overline{c}_1, \dots, \overline{c}_s$ are completely representable.

for $i = 1, \dots, s$ **do**
  $\overline{b} := \overline{c}_i$, $\overline{b}^{(k-1)} := \overline{b}^{[k-1]}$ (the designated witness to $\overline{b}|_{[k-1]} = \overline{c}_i|_{[k-1]} \in B|_{[k-1]}$)
  for $m = k, \dots, n$ **do**
    $\beta := \overline{b}^{(m-1)}|_m$, $\gamma := \overline{b}|_m$, $\overline{c} := p(\overline{b}^{(m-1)}, \overline{b}, \overline{b})$ (p as defined in Lemma 5.2.5)
    if R has no designated witnesses for $(\gamma, \beta^\gamma) \in \mathsf{FORK}_m(R)$, **then**
      add $\overline{b}, \overline{c}$ to R as designated witnesses for $(\gamma, \beta^\gamma) \in \mathsf{FORK}_m(R)$
    endif
    if R has no designated witnesses for $(\gamma, \beta) \in \mathsf{FORK}_m(R)$, **then**
      add $\overline{b}, \overline{b}^{(m-1)}$ to R as designated witnesses for $(\gamma, \beta) \in \mathsf{FORK}_m(R)$
    endif
    Let $\overline{u}, \overline{v} \in R$ be the designated witnesses for $(\gamma, \beta^\gamma) \in \mathsf{FORK}_m(R)$
    $\overline{b}^{(m)} := t_m(\overline{b}^{(m-1)}, \overline{v}, \overline{u}, (\overline{b^J})_{J \in \binom{[m]}{k-1}})$ where $(\overline{b^J})_{J \in \binom{[m]}{k-1}}$ is a tuple of designated local witnesses from R (so that $\overline{b^J}|_J = \overline{b}|_J$)
  endfor
endfor
return $R' := R$

The correctness of Algorithm 2 follows from the definitions involved together with Lemma 5.2.9 and Lemma 5.2.11. Indeed correctness here is essentially a porism of Lemma 5.2.11.

[10] Recall that the input $\mathbf{C}$ includes the operation table of each basic operation of $\mathbf{C}$ and hence if f is an l-ary basic operation of $\mathbf{C}$, then $||\mathbf{C}|| \geq |C|^l$. In particular, if $\mathbf{C}$ has an l-ary basic operation, then any function which is $O(|C^J|^l)$ is therefore $O(||\mathbf{C}^J||)$.

[11] Formally the sets A^I and $A^{|I|}$ are different sets but up to a suitable encoding they are essentially the same.

For the complexity, notice that the outermost **for** loop is executed s times. In Step 2 we need to read and copy the tuple $\bar{c}_i$ and then lookup and copy the designated witness in R. Assuming a constant lookup time (since the table for R is given in the input) this is achieved in time $O(n)$. The inner for loop is run $n-k$ times. It remains to calculate the time required to complete Steps 4-12 and 15.

Step 4 requires evaluating the term operation p for a tuple of inputs of length n. Hence a time requirement here of $O(n)$. Checking the conditions of the **if** statements in Steps 5 and 8 both require $O(||R||) = O(n^{k-1}||\mathbf{C}||^{k-1})$ time. This clearly dominates the time taken to add the correct tuples to R in Steps 6 and 9, and Step 11 can be achieved simultaneously with either Step 5 or Step 6 (depending on whether or not the witnesses were already included in R). The estimate found in Lemma 5.2.9 for the time requirement of Step 12 is $O(m^k)$ which is $O(n^k)$ since $m \leq n$.

Step 15 requires time $O(||R'||) = O(n^{k-1}||\mathbf{C}||^{k-1})$.

Hence the runtime of Algorithm 2 is:

$$O(s(n + (n-k)(n + n^{k-1}||\mathbf{C}||^{k-1} + n^k)) + n^{k-1}||\mathbf{C}||^{k-1})$$

which is $O(sn^k||\mathbf{C}||^{k-1} + sn^{k+1})$, a polynomial in n and $||\mathbf{C}||$.

Our first use of Algorithm 2 is to extend the PSR R of $\langle \pi_1, \ldots, \pi_r \rangle_{\mathbf{A}^I}$ constructed after running Algorithm 1 on the input $\mathbf{A}, \pi_1, \ldots, \pi_r$. This has a time requirement of

$$O(r(|A|^r)^k||\mathbf{A}||^{k-1} + r(|A|^r)^{k+1})$$

(using the estimate $|I| \leq |A|^r$) which is $O(||\mathbf{A}||^{k(1+r)-1} + ||\mathbf{A}||^{r(k+1)})$.

So far we have achieved the following in our proof of Theorem 5.2.4:

- Construct the table of a $(1, k-1)$-parallelogram term operation of the algebra $\mathbf{A}$ in time $O(||\mathbf{A}||^{4k+3})$,
- Construct an instance $g, \pi_1, \ldots, \pi_r$ of SMP($\mathbf{A}$) which is equivalent to the instance $\mathbf{A}$ of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ in time $O(||\mathbf{A}||^r)$,
- Construct a PSR R of $\langle \pi_1, \ldots, \pi_r \rangle_{\mathbf{A}^I}$ with a full set of designated local witnesses in time $O(||\mathbf{A}||^{k(1+r)})$,
- Extend the PSR R to a PSR R' with respect to which each π_i is completely representable in time $O(||\mathbf{A}||^{k(1+r)-1} + ||\mathbf{A}||^{r(k+1)})$.

Each of the items accomplished so far have required only polynomial-time in the size of the input $||\mathbf{A}||$. Next we will introduce the NP oracle defined in the proof of [13, Theorem 4.12] which will give us a polynomial-time verifiable certificate for $g \in \langle \pi_1, \ldots, \pi_r \rangle_{A^I}$. We reproduce the oracle as seen in that paper, analyze the complexity of verifying a "YES" output of the oracle and then describe precisely the certificate for "$\mathbf{A}$ is a YES instance of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$".

Oracle: Need More Fork Witnesses
Input: A PSR R of $\mathbf{R}^* := \langle R \rangle_{A^n}$ with a full set of designated local witnesses.
Output:

- (YES, $h, (\overline{u}_j^{(k)}, \overline{v}_j^{(k)}, \ldots, \overline{u}_j^{(n)}, \overline{v}_j^{(n)}, \overline{w_j})_{j=1}^{l}, \overline{b}_1, \ldots, \overline{b}_l, \overline{b}, R'$) where:
  1. h is the operation symbol of an l-ary basic operation of $\mathbf{A}$,[12]
  2. Each pair $(\overline{u}_j^{(m)}, \overline{v}_j^{(m)})$ are designated witnesses to a fork in $\mathsf{FORK}_m(R)$,
  3. Each $\overline{w_j} = (\overline{w_j^J})_{J \in \binom{[n]}{k-1}}$ is a tuple of designated local witnesses to $\overline{b}_j|_J \in R|_J$,
  4. $\overline{b}_j = T_n(\overline{u}_j^{(k)}, \overline{v}_j^{(k)}, \ldots, \overline{u}_j^{(n)}, \overline{v}_j^{(n)}, (\overline{w_j^J})_{J \in \binom{[n]}{k-1}})$,
  5. $\overline{b} = h^{\mathbf{A}}(\overline{b}_1, \ldots, \overline{b}_l)$,
  6. $\overline{b}$ is *not* completely representable by R,
  7. $R' \supseteq R$ is the output of **Algorithm 2** run on the input $\mathbf{A}, P, \overline{b}, R$ (so that $\overline{b}$ *is* completely representable by R');
- NO, if $h, (\overline{u}_j^{(k)}, \overline{v}_j^{(k)}, \ldots, \overline{u}_j^{(n)}, \overline{v}_j^{(n)}, \overline{w_j})_{j=1}^{l}, \overline{b}_1, \ldots, \overline{b}_l, \overline{b}$ with these properties do not exist.

We now show that given a PSR R of $\mathbf{R}^*$ with a full set of designated local witnesses and given the output (YES, $h, (\overline{u}_j^{(k)}, \overline{v}_j^{(k)}, \ldots, \overline{u}_j^{(n)}, \overline{v}_j^{(n)}, \overline{w_j})_{j=1}^{l}, \overline{b}_1, \ldots, \overline{b}_l, \overline{b}, R'$) we can verify the seven conditions listed in polynomial-time (with respect to n and $||\mathbf{A}||$).

Condition 1 can clearly be checked in time $O(||\mathbf{A}||)$. Conditions 2 and 3 rely on searching the input PSR R, which can be achieved in time $O(||R||)$ which is $O(n^{k-1}||\mathbf{A}||^{k-1})$. For condition 2 there are $l(n-k)$ different tuples to look for, while condition 3 requires $l\binom{n}{k-1}$ searches of the PSR R. Hence condition 2 is checked in time $O(ln^k||\mathbf{A}||^{k-1})$ and condition 3 is checked in time $O(ln^{2k-2}||\mathbf{A}||^{k-1})$. Using the analysis preceding Lemma 5.2.11, condition 4 can be checked in time $O(ln^{k+1})$. Condition 5 can be checked using n table lookups in the table for $h^{\mathbf{A}}$, each of which is a constant-time procedure, hence $O(n)$.

Conditions 6 and 7 can be checked in time $O(n^k||\mathbf{A}||^{k-1} + n^{k+1})$ by running Algorithm 2 on the input $\mathbf{A}, P, \overline{b}, R$ (and noting that the output R' is distinct from the input R). Since R contains designated local witnesses for $\mathbf{R}^*$ and $\overline{b} = h^{\mathbf{A}}(\overline{b}_1, \ldots, \overline{b}_l) \in \mathbf{R}^*$, this is a valid input for Algorithm 2.

Hence correctness of the output

$$(\text{YES}, h, (\overline{u}_j^{(k)}, \overline{v}_j^{(k)}, \ldots, \overline{u}_j^{(n)}, \overline{v}_j^{(n)}, \overline{w_j})_{j=1}^{l}, \overline{b}_1, \ldots, \overline{b}_l, \overline{b}, R')$$

can be verified in time

$$O(||\mathbf{A}|| + ln^k||\mathbf{A}||^{k-1} + ln^{2k-2}||\mathbf{A}||^{k-1} + ln^{k+1} + n + n^k||\mathbf{A}||^{k-1} + n^{k+1})$$

which can be simplified to $O(ln^{2k-2}||\mathbf{A}||^{k-1})$ which is $O(n^{2k-2}||\mathbf{A}||^k)$ where we have used the (extremely crude) approximation $l \leq ||\mathbf{A}||$.

[12]Depending on the encoding of the algebra $\mathbf{A}$, h is a suitable index pointing to a specific basic operation of $\mathbf{A}$.

We are now finally ready to describe a polynomial-time verifiable certificate for the statement "$\mathbf{A}$ is a YES instance of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$". Suppose that $\mathbf{A}$ is a YES instance of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$. Let $g, \pi_1, \ldots, \pi_r$ be the corresponding YES instance of SMP($\mathbf{A}$) constructed earlier in the proof and let S be the PSR of $\langle \pi_1, \ldots, \pi_r \rangle_{A^I}$ constructed after running Algorithms 1 and 2 on the appropriate inputs, as described earlier in the proof. Notice that the PSR S satisfies the necessary assumption to be inputted into the oracle Need More Fork Witnesses, namely, S has designated local witnesses for $\mathbf{S}^*$.[13] Notice also that for any valid input R to the oracle Need More Fork Witnesses, given the output $(\text{YES}, h, (\overline{u}_j^{(k)}, \overline{v}_j^{(k)}, \ldots, \overline{u}_j^{(n)}, \overline{v}_j^{(n)}, \overline{w_j})_{j=1}^{l}, \overline{b}_1, \ldots, \overline{b}_l, \overline{b}, R')$, it also follows that R' is a valid input to the same oracle (since $\langle R' \rangle = \langle R \rangle$ and $R \subseteq R'$).

There are two possible outcomes after running the oracle Need More Fork Witnesses on the input S:

- The oracle returns something of the form

$$(\text{YES}, h, (\overline{u}_j^{(k)}, \overline{v}_j^{(k)}, \ldots, \overline{u}_j^{(n)}, \overline{v}_j^{(n)}, \overline{w_j})_{j=1}^{l}, \overline{b}_1, \ldots, \overline{b}_l, \overline{b}, S')$$

  satisfying conditions (1)-(7) [and hence $\overline{b} \in \langle \pi_1, \ldots, \pi_r \rangle$ is not completely representable by S], or

- Every element of $\mathbf{S}^* = \langle \pi_1, \ldots, \pi_r \rangle$ is completely representable by S (in which case the oracle returns NO).

If there is some element of $\mathbf{S}^*$ which is not completely representable by S, then we let Y_1 be the string $(\text{YES}, h, (\overline{u}_j^{(k)}, \overline{v}_j^{(k)}, \ldots, \overline{u}_j^{(n)}, \overline{v}_j^{(n)}, \overline{w_j})_{j=1}^{l}, \overline{b}_1, \ldots, \overline{b}_l, \overline{b}, S')$ outputted by the oracle and then call the oracle again on the output $S^{(1)} := S'$. We again have two possible outcomes and can inductively define $S^{(i)} := (S^{(i-1)})'$ as long as the oracle returns something of the form

$$(\text{YES}, h, (\overline{u}_j^{(k)}, \overline{v}_j^{(k)}, \ldots, \overline{u}_j^{(n)}, \overline{v}_j^{(n)}, \overline{w_j})_{j=1}^{l}, \overline{b}_1, \ldots, \overline{b}_l, \overline{b}, (S^{(i-1)})')$$

on the input $S^{(i-1)}$, defining Y_i similarly as the output obtained after calling the oracle on the input $S^{(i-1)}$.

What we obtain is a sequence of strings $Y_1, Y_2, \ldots, Y_K$ of "YES" outputs from the oracle corresponding to input PSRs $S^{(0)} := S \subset S^{(1)} \subset \cdots \subset S^{(K-1)}$. Each $S^{(i)}$ is a proper subset of $S^{(i+1)}$ and they are all PSRs of the subpower $\langle \pi_1, \ldots, \pi_r \rangle_{A^I}$. Indeed, each $S^{(i+1)}$ is obtained from $S^{(i)}$ by adding at least one new pair of designated fork witnesses witnessing forks in one of the coordinates $k, \ldots, n = |I|$. In each coordinate there are a maximum of $|A|^2$-many forks and hence the number K of "YES" outputs that we obtain before obtaining the first output of "NO" is at most $|A|^2(|I| - k)$ which is $O(||\mathbf{A}||^{r+2})$.

Given the certificate $Y_1, \ldots, Y_K$ we can verify $g \in \mathbf{B} := \langle \pi_1, \ldots, \pi_r \rangle$ as follows:

[13] Every element of S is in the subalgebra $\langle \pi_1, \ldots, \pi_r \rangle_{A^I}$ by construction, and hence $\mathbf{S}^* = \langle \pi_1, \ldots, \pi_r \rangle$ and therefore Algorithm 1 guarantees that S satisfies this assumption.

- Verify that g is completely representable by $S^{(K)}$. I.e. verify that:

  1. for every choice of indices $J \in \binom{[|I|]}{k-1}$, $S^{(K)}$ contains an element g^J which is a designated local witness for $g|_J \in \mathbf{B}|_J$, and
  2. for every $k \leq m \leq |I|$, $S^{(K)}$ contains elements $\overline{u}^{(m)}, \overline{v}^{(m)}$ designated to witness the fork $(g|_m, \beta^{g|_m})$ where

  $$\beta = T_{m-1}(\overline{u}^{(k)}, \overline{v}^{(k)}, \ldots, \overline{u}^{(m-1)}, \overline{v}^{(m-1)}, (g^J)_{J \in \binom{[m-1]}{k-1}})|_m.$$

  Achieving Item 1 involves searching in the PSR $S^{(K)}$ $O(|I|^{k-1})$-times and can therefore be achieved in time

  $$O(||S^{(K)}|||A|^{r(k-1)}) = O((|A|^r)^{k-1}||\mathbf{A}||^{k-1}(|A|^r)^{k-1})$$

  which is

  $$O(||\mathbf{A}||^{(2r+1)(k-1)}).$$

  In Item 2 we search the PSR $S^{(K)}$ $(|I|-k)$-times after first calculating $(|I|-k)$ values of T_{m-1}. This takes time

  $$O(||S^{(K)}|||I| + |I|(m-1)^{(k+1)}) = O((|A|^r)^{k-1}||\mathbf{A}||^{k-1}|A|^r + |A|^r(|A|^r)^{k+1})$$

  which is

  $$O(||\mathbf{A}||^{rk+k-1} + ||\mathbf{A}||^{rk+r+1}).$$

- For each $i = 2, \ldots, K$, verify that Y_i is a valid output of the oracle Need More Fork Witnesses on the input $S^{(i-1)}$. Each verification is achieved in time

  $$O(|I|^{2k-2}||\mathbf{A}||^k) = O((|A|^r)^{2k-2}||\mathbf{A}||^k) = O(||\mathbf{A}||^{2kr+k-2r})$$

  and there are $O(||\mathbf{A}||^{r+2})$ many Y_i to verify for a time requirement of

  $$O(||\mathbf{A}||^{r+2}||\mathbf{A}||^{2kr+k-2r}) = O(||\mathbf{A}||^{2kr+k-r+2}).$$

- Verify that Y_1 is a valid output of the oracle Need More Fork Witnesses on the input S which is obtained by running Algorithm 1 on the input $\mathbf{A}$, $\pi_1, \ldots, \pi_r$ to get the output R and then running Algorithm 2 on the input $\mathbf{A}$, P, $\pi_1, \ldots, \pi_r$, R. This is achieved in time

  $$O(||\mathbf{A}||^{k(1+r)} + ||\mathbf{A}||^{k(1+r)-1} + ||\mathbf{A}||^{r(k+1)}) = O(||\mathbf{A}||^{k(1+r)+r(k+1)}).$$

Since we have demonstrated that this procedure entails verification of polynomially-many claims, each of which we have demonstrated can be checked in polynomial-time, it follows that $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ is in NP, as required.[14]

□

[14]Since k and r are both at least 2, the bounds obtained for the verifier can be replaced by the cruder but simpler estimate $O(||\mathbf{A}||^{3kr})$ when this is more convenient.

We end this chapter by noting that for many well-known conditions of height < 1 the idempotent Mal'tsev condition satisfaction problem is actually tractable. Examples include Mal'tsev terms [20], k-ary near unanimity terms (for fixed k) [24], and k-edge terms (for fixed k)[24]. Perhaps one of the most interesting cases is that of the minority term for which it is still unknown whether a polynomial-time algorithm exists. The result of [29, Theorem 17] placed this problem in the complexity class NP and we are pleased now to have extended this to all conditions of height < 1. We pose the following questions as suggestions for further study:

Question 5.2.13.

- *What is the complexity of the problem* $Sat^{Id}_{Minority}$*?*
- *Are there Mal'tsev conditions* Σ *of height* < 1 *such that* Sat^{Id}_{Σ} *is NP-complete?*
- *Are there Mal'tsev conditions* Σ *of height* < 1 *such that* Sat^{Id}_{Σ} *is NP-intermediate (assuming* $P \neq NP$*)?*
- *If there are no Mal'tsev conditions* Σ *of height* < 1 *with NP-intermediate idempotent MCSPs, then is there a "nice" characterization of which problems are in P and which are NP-complete? Here we imagine something akin to the dichotomy theorem for CSPs [14], [44].*

In the next (and final) chapter, we summarize the results obtained in this thesis and provide a common generalization to Theorems 5.2.4 and 3.2.1.

Chapter 6

Conclusion

We conclude the thesis by summarizing for convenience the new results obtained herein and then providing a list of open questions as suggestions for further research.

6.1 Summary of Results

We obtain the following tractability results.

Theorem 6.1.1 (3.1.8)**.** *The problem Sat^{Id}_{Σ} is in the complexity class P if Σ is a linear strong Mal'tsev condition which implies $\mathcal{NU}$.*

Theorem 6.1.2. *The "search problem" whose input is a finite idempotent algebra $\mathbf{A}$ and whose output is a collection of term operations of $\mathbf{A}$ which satisfy the equations of the strong Mal'tsev condition Σ (or the answer "NO" if such terms do not exist) is in P whenever Σ is*

- *(4.2.1) existence of an idempotent cyclic term*
- *(4.4.5) a Mal'tsev condition of height < 1 satisfying the downward column condition*
- *(4.5.1) (for some fixed $n \geq 2$) existence of a sequence of n Hagemann-Mitschke terms*

We also prove that two problems are in NP.

Theorem 6.1.3. *The problem Sat^{Id}_{Σ} is in NP when Σ is a strong Mal'tsev condition which*

- *(3.2.1) implies $\mathcal{NU}$, or*
- *(5.2.4) is of height < 1.*

By examining the proofs of Theorems 3.2.1 and 5.2.4 we arrive at the pleasing common generalization.

Theorem 6.1.4. *Let Σ be a strong Mal'tsev condition which implies the existence of an edge term. Then Sat^{Id}_{Σ} is in NP.*

Sketch of Proof. Firstly note that a result similar to Lemma 3.1.3 can be obtained to show that if Σ implies the existence of an edge term, then Σ implies the existence of a k-edge term (and hence a $(1, k-1)$-parallelogram term [32, Theorem 3.5]) for some fixed $k \geq 2$.

Given a finite idempotent algebra $\mathbf{A}$ the first step then is to use the result of Proposition 5.2.3 to obtain the operation table of a $(1, k-1)$-parallelogram term operation of $\mathbf{A}$ if such a term operation exists (and otherwise return the answer that $\mathbf{A}$ does not satisfy Σ).

In the proof of Theorem 5.2.4 we use the equations of Σ (which are assumed in that case to be of height < 1) to construct partial operations which satisfy those equations. Clearly such a procedure cannot be carried out when Σ contains equations which are not of height < 1. However if the algebra $\mathbf{A}$ is a YES instance of $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$ then we may include in the certificate not only the string $Y_1, \ldots, Y_K$ obtained in the proof of Theorem 5.2.4 but also tables for the term operations $g_1, \ldots, g_m$ obtained in the proof of Theorem 3.2.1 which witness that $\mathbf{A}$ satisfies Σ. The proof of Theorem 3.2.1 outlines how to check that the operations $g_1, \ldots, g_m$ satisfy the requisite equations in time $||\mathbf{A}||^P$ where P depends only on Σ and not on the instance $\mathbf{A}$. The proof of Theorem 5.2.4 outlines how to verify that the operations $g_1, \ldots, g_m$ are term operations of the algebra $\mathbf{A}$ by setting up suitable instances of $\mathrm{SMP}(\mathbf{A})$. This verification can be achieved in time $O(||\mathbf{A}||^{3kR})$ where $(k+3)$ is the arity of the $(1, k-1)$-parallelogram term guaranteed by Σ and R is the maximum arity of each of the operations $g_1, \ldots, g_m$, both of which are again independent of the algebra $\mathbf{A}$. Hence there is a polynomial-time verifier for $\mathrm{Sat}^{\mathrm{Id}}_{\Sigma}$, as required. □

Corollary 6.1.5. *If the search problem related to $Sat_{Edge(k)}$ is in NP then so is the decision problem (and the related search problem for) Sat_{Σ} for any Σ which implies the existence of a k-edge term.*

6.2 Open Questions

We now bring together those natural questions arising as potential directions for further study.

Question 6.2.1. *What is the complexity of $Sat_{\mathcal{NU}}$? What about $Sat_{\mathcal{NU}(k)}$ for fixed $k \geq 3$?*

The obvious brute force algorithm for $\mathrm{Sat}_{\mathcal{NU}(k)}$ (build all k-ary term operations and check whether any is a near unanimity term) runs in exponential time but in

light of Corollaries 3.1.10 and 3.2.2 any improvement on this algorithm will lead to corresponding improvements for the problem Sat_Σ when Σ implies $\mathcal{NU}$.

The following related problems also arise naturally from considering the results found in Chapter 3 of this thesis.

Question 6.2.2. *What is the complexity of Sat^{Id}_Λ, where Λ is the strong Mal'tsev condition of having lattice terms?*

Question 6.2.3. *What is the complexity of $Sat^{Id}_{2\text{-}semi}$, where 2-semi is the Mal'tsev condition of having a 2-semilattice term?*

In Chapter 3 we describe the ethos that led us to conjecture that the first of these problems ought to be sub-exponential while the second ought to be EXPTIME-complete but as yet there is only limited evidence for these conjectures.

In Chapter 5 our considerations led us to pose the following questions (appearing there as part of Question 5.2.13).

Question 6.2.4.

- *Are there Mal'tsev conditions Σ of height < 1 such that Sat^{Id}_Σ is NP-complete?*
- *Are there Mal'tsev conditions Σ of height < 1 such that Sat^{Id}_Σ is NP-intermediate (assuming $P \neq NP$)?*
- *If there are no Mal'tsev conditions Σ of height < 1 with NP-intermediate idempotent MCSPs, then is there a "nice" characterization of which problems are in P and which are NP-complete? Here we imagine something akin to the dichotomy theorem for CSPs [14], [44].*

We also pose the following question which has certainly been asked before (see for example [29]).

Question 6.2.5. *What is the complexity of $Sat^{Id}_{Minority}$, where Minority is the Mal'tsev condition of having a minority term?*

In the paper [29] it is shown that the condition Minority does not have the idempotent local-global property (Definition 4.4.8). It may be that other techniques can be used to determine a polynomial-time algorithm for $\mathrm{Sat}^{\mathrm{Id}}_{\mathrm{Minority}}$. If this is the case, it would provide an answer to the following more general question posed in Chapter 4.

Question 6.2.6. *Is there a strong Mal'tsev condition whose (idempotent) satisfaction problem is in P but which does not have the (idempotent) local-global property?*

The final questions raised in this thesis all focus on the related search problem for Mal'tsev condition satisfaction.

Question 6.2.7. *If Σ is a Mal'tsev condition arising as $M(P)$ for a pattern graph P as outlined in [30, Section 3.2], does it follow that there is a polynomial-time algorithm which takes as input a finite idempotent algebra $\mathbf{A}$ and returns term operations of $\mathbf{A}$ which witness that $\mathbf{A}$ satisfies the Mal'tsev condition Σ whenever such terms exist (and otherwise returns the answer NO)?*

Question 6.2.8. *Is there a strong Mal'tsev condition which has the local-global property (Definition 4.4.8) but for which the corresponding search problem of obtaining term operations witnessing the satisfaction is not in P?*

Question 6.2.9. *More generally, is there a strong Mal'tsev condition whose satisfaction problem is in P but for which the corresponding search problem of obtaining term operations is not in P?*

If pushed, we would conjecture that the answers to these last three questions are respectively "Yes", "No", and "No" but we leave it to future research to determine the correctness of these guesses.

There are certainly many more questions arising in the field of Mal'tsev condition satisfaction and those outlined here represent only a select few of particular relevance to the framing of this thesis in this author's mind. There is clearly a long way to travel before we have this region of complexity theory completely mapped out and we invite the interested reader to dive in to the references that follow for further exploration.

Bibliography

[1] Erhard Aichinger. Constantive Mal'cev clones on finite sets are finitely related. *Proc. Amer. Math. Soc.*, 138(10):3501–3507, 2010.

[2] Kirby A. Baker and Alden F. Pixley. Polynomial interpolation and the Chinese remainder theorem for algebraic systems. *Math. Z.*, 143(2):165–174, 1975.

[3] Libor Barto and Marcin Kozik. Absorbing subalgebras, cyclic terms, and the constraint satisfaction problem. *Log. Methods Comput. Sci.*, 8(1:07):1–26, 2012.

[4] Libor Barto, Marcin Kozik, Miklós Maróti, Ralph McKenzie, and Todd Niven. Congruence modularity implies cyclic terms for finite algebras. *Algebra Universalis*, 61(3):365–380, 2009.

[5] Libor Barto, Andrei Krokhin, and Ross Willard. Polymorphisms, and how to use them. In *The constraint satisfaction problem: complexity and approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 1–44. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2017.

[6] Libor Barto, Jakub Opršal, and Michael Pinsker. The wonderland of reflections. *Israel J. Math.*, 223(1):363–398, 2018.

[7] Clifford Bergman. *Universal algebra*, volume 301 of *Pure and Applied Mathematics (Boca Raton)*. CRC Press, Boca Raton, FL, 2012. Fundamentals and selected topics.

[8] Joel Berman, Paweł Idziak, Petar Marković, Ralph McKenzie, Matthew Valeriote, and Ross Willard. Varieties with few subalgebras of powers. *Trans. Amer. Math. Soc.*, 362(3):1445–1473, 2010.

[9] Garrett Birkhoff. On the structure of abstract algebras. *Proc. Cambr. Philos. Soc.*, 31(4):433–454, 1935.

[10] Andrei Bulatov. Combinatorial problems raised from 2-semilattices. *Journal of Algebra*, 298:321–339, 2006.

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

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

[13] Andrei Bulatov, Peter Mayr, and Ágnes Szendrei. The subpower membership problem for finite algebras with cube terms. *Log. Methods Comput. Sci.*, 15(1):Paper No. 11, 48, 2019.

[14] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *58th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2017*, pages 319–330. IEEE Computer Soc., Los Alamitos, CA, 2017.

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

[16] Hubie Chen and Benoit Larose. Asking the metaquestions in constraint tractability. *ACM TOCT*, 9(3):11:1–11:27, 2017.

[17] Tomás Feder and Moshe Y. Vardi. Monotone monadic snp and constraint satisfaction. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 612–622, New York, NY, USA, 1993. ACM Press.

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

[19] Ralph Freese, J. B. Nation, and Matt Valeriote. Testing for a semilattice term. *Order*, 36(1):65–76, 2019.

[20] Ralph Freese and Matthew A. Valeriote. On the complexity of some Maltsev conditions. *Internat. J. Algebra Comput.*, 19(1):41–77, 2009.

[21] Joachim Hagemann and A. Mitschke. On n-permutable congruences. *Algebra Universalis*, 3:8–12, 1973.

[22] David Hobby and Ralph McKenzie. *The structure of finite algebras*, volume 76 of *Contemporary Mathematics*. American Mathematical Society, Providence, RI, 1988. Revised edition: 1996.

[23] Jonah Horowitz. *Results on the Computational Complexity of Linear Idempotent Mal'cev Conditions*. PhD thesis, McMaster University, 2012.

[24] Jonah Horowitz. Computational complexity of various Mal'cev conditions. *Internat. J. Algebra Comput.*, 23(6):1521–1531, 2013.

[25] Paweł Idziak, Petar Marković, Ralph McKenzie, Matthew Valeriote, and Ross Willard. Tractability and learnability arising from algebras with few subpowers.

In *LICS '07: Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science*, pages 213–224, Washington, DC, USA, 2007. IEEE Computer Society.

[26] Paweł M. Idziak and Jacek Krzaczkowski. Satisfiability in multi-valued circuits. In *LICS '18—33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, page [9 pp.]. ACM, New York, 2018.

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

[28] Peter Jeavons, David Cohen, and Martin C. Cooper. Constraints, consistency and closure. *Artificial Intelligence*, 101(1-2):251–265, 1998.

[29] Alexandr Kazda, Jakub Opršal, Matt Valeriote, and Dmitriy Zhuk. Deciding the existence of minority terms. *Accepted by the Canadian Bulletin of Mathematics*, 2019.

[30] Alexandr Kazda and Matt Valeriote. Deciding some Maltsev conditions in finite idempotent algebras. *Accepted by the Journal of Symbolic Logic*, 2019.

[31] Keith A. Kearnes and Emil W. Kiss. The shape of congruence lattices. *Mem. Amer. Math. Soc.*, 222(1046):viii+169, 2013.

[32] Keith A. Kearnes and Ágnes Szendrei. Clones of algebras with parallelogram terms. *Internat. J. Algebra Comput.*, 22(1):1250005, 30, 2012.

[33] Richard E. Ladner. On the structure of polynomial time reducibility. *J. Assoc. Comput. Mach.*, 22:155–171, 1975.

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

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

[36] Peter Mayr. The subpower membership problem for Mal′cev algebras. *Internat. J. Algebra Comput.*, 22(7):1250075, 23, 2012.

[37] George F. McNulty. Undecidable properties of finite sets of equations. *J. Symbolic Logic*, 41(3):589–604, 1976.

[38] 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.

[39] Mark H. Siggers. A strong Mal'cev condition for locally finite varieties omitting the unary type. *Algebra Universalis*, 64(1-2):15–20, 2010.

[40] Michael Sipser. *Introduction to the Theory of Computation*. Course Technology, second edition, 2006.

[41] Walter Taylor. Simple equations on real intervals. *Algebra Universalis*, 61(2):213–226, 2009.

[42] M. Valeriote and R. Willard. Idempotent n-permutable varieties. *Bull. Lond. Math. Soc.*, 46(4):870–880, 2014.

[43] Matt Valeriote and Ross Willard. Private Communication from Dr Matt Valeriote, Dept. of Mathematics and Statistics, McMaster University.

[44] Dmitriy Zhuk. A proof of CSP dichotomy conjecture. In *58th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2017*, pages 331–342. IEEE Computer Soc., Los Alamitos, CA, 2017.