1-1-2019

Decidability for Residuated Lattices and Substructural Logics

Gavin St. John
University of Denver

Decidability for Residuated Lattices and Substructural Logics

Abstract

We present a number of results related to the decidability and undecidability of various varieties of residuated lattices and their corresponding substructural logics. The context of this analysis is the extension of residuated lattices by various simple equations, dually, the extension of substructural logics by simple structural rules, with the aim of classifying simple equations by the decidability properties shared by their extensions. We also prove a number of relationships among simple extensions by showing the equational theory of their idempotent semiring reducts coincides with simple extensions of idempotent semirings. On the decidability front, we develop both semantical and syntactical methods for establishing decidability as well as tractability of decision procedures. On the undecidability front, we develop a notion of algebraic machines for which the theory of residuated frames will allow us to encode decision problems within the theories of residuated lattices and their substructural analogues. We prove the undecidability of the word problem for a broad class of simple extensions for both commutative and non-commutative residuated lattices. Furthermore, through a deduction theorem we establish the undecidability of the equational theory for a broad class of simple extensions. Translated in terms of substructural logics, we prove that the undecidability of both provability and deducibility for a multitude of extensions of FLe by simple rules.

Document Type

Dissertation

Degree Name

Ph.D.

Department

Mathematics

First Advisor

Nikolaos Galatos, Ph.D.

Keywords

Algebraic logic, Decidability, Residuated lattice, Substructural logic, Undecidability

Subject Categories

Algebra | Mathematics | Physical Sciences and Mathematics

Decidability for residuated lattices and substructural logics

---

A Dissertation

Presented to

the Faculty of Natural Sciences and Mathematics

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

---

by

Gavin St. John

June 2019

Advisor: Nikolaos Galatos

Author: Gavin St. John
Title: Decidability for residuated lattices and substructural logics
Advisor: Nikolaos Galatos
Degree Date: June 2019

ABSTRACT

We present a number of results related to the decidability and undecidability of various varieties of residuated lattices and their corresponding substructural logics. The context of this analysis is the extension of residuated lattices by various simple equations, dually, the extension of substructural logics by simple structural rules, with the aim of classifying simple equations by the decidability properties shared by their extensions. We also prove a number of relationships among simple extensions by showing the equational theory of their idempotent semiring reducts coincides with simple extensions of idempotent semirings. On the decidability front, we develop both semantical and syntactical methods for establishing decidability as well as tractability of decision procedures. On the undecidability front, we develop a notion of algebraic machines for which the theory of residuated frames will allow us to encode decision problems within the theories of residuated lattices and their substructural analogues. We prove the undecidability of the word problem for a broad class of simple extensions for both commutative and non-commutative residuated lattices. Furthermore, through a deduction theorem we establish the undecidability of the equational theory for a broad class of simple extensions. Translated in terms of substructural logics, we prove that the undecidability of both provability and deducibility for a multitude of extensions of $\mathbf{FL}_{\mathrm{e}}$ by simple rules.

ACKNOWLEDGEMENTS

First and foremost, I must thank my advisor Nick Galatos. Nick has the unique ability to both inspire his students with his brilliance and empower them with his optimism and encouragement. Nick makes mathematics exciting, and I am grateful for every chance I have had to explore the mathematical world with him. Without his mentoring, collaboration, and zealous character, I would be lost.

I am deeply grateful to Dr. Wesley Fussner, my academic brother. Our camaraderie during graduate school was essential to my development as a mathematician. In particular, I thank Wesley for sharing with me his intuitions and helping me cultivate my understanding of logic. I am indebted to Dr. Sara Ugolini, who has been my mathematical muse. This dissertation could not have been completed without her support and guidance. I sincerely thank Wesley and Sara for their inspiration as researchers, as mathematicians, and most importantly, as friends.

I would like to thank the many friends and mentors who enabled my journey into higher academia. My fascination with reason truly began at the University of Pittsburgh with a group composed of Vincent Isaac Page, Alex Pruszenski, and Katherine "Kitty" Durgin. Our many coffee-fueled nights discussing mathematics, physics, and philosophy serve as the bedrock for my inspiration to study mathematical ideas. I am grateful to Dr. L.J. Nickisch and the rest of the team at NorthWest Research Associates for their encouragement and guidance. I also must thank the many researchers I have met during my time at Youngstown State University, in particular, Dr. Nathan Ritchey, Dr. Stephen E. Rodabaugh, Dr. Jeffrey T. Denniston, Dr. Jamal Tartir, Dr. Zbigniew Piotrowski, and Dr. Neil Flowers. I would also like to thank the entire mathematics department at the University of Denver, in particular Dr. Petr Vojtechovsky and Dr. Michael Kinyon.

Lastly, I extend my deepest gratitude to my parents Lori and Timothy and my siblings Brittany and Tyler, whose love and support are the foundation I rest upon in all my pursuits.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: Introduction

Decidability is a fundamental problem in the study of mathematical logic. In short, a logic is *decidable* if there exists an algorithm for determining whether or not any given formula is provable. Classical propositional logic is so explicitly decidable that we teach it to college sophomores when they learn truth tables. On the other hand, first-order classical logic is undecidable as consequence of Gödel's incompleteness theorems, that is, there *cannot in principle* exist an algorithm for determining provability.

As many mathematical fields are rooted in the investigation of certain first-order theories of classical logic, a distinction between what is *true* and what is *provable*, semantics and syntax, was made. Different concepts of truth and provability arose, giving birth to the formulation and study of *nonclassical logics*, which can be viewed as any departure from the classical setting.

In particular, a framework that includes most of the interesting nonclassical logics is given by *substructural logics*. Substructural logics encompass, besides classical logic, intuitionistic logic, relevance logics, many-valued logics, fuzzy logics, linear logic and their non-commutative versions. Originally, substructural logics were introduced as logics which, when formulated as Gentzen-style systems, lack some (including "none" as a special case) of the three basic structural rules for intuitionistic logic, *contraction* (c)*, weakening* (w) and *exchange* (e). For example, relevance logics and linear logic lack the weakening rule, many-valued logics, fuzzy logics and linear logic lack the contraction rule, and hence all of them can be regarded as substructural logics. The Gentzen system for intuitionistic logic $\mathbf{LJ}$ is equivalently denoted $\mathbf{FL}_{\mathsf{ecw}}$ as a structural extension of the *Full-Lambek calculus* $\mathbf{FL}$.

A powerful tool for analyzing substructural logics uniformly is given by semantical methods, due to the fact that they are *algebraizable*. Indeed, syntactic properties of algebraizable logics can be rendered as semantical properties for a particular variety of algebras, and *vice versa*. In particular, decidability properties of a logic can be handled abstractly in the algebraic setting.

The algebraic models of substructural logics are *residuated lattices*. Residuated lattices encompass a broad class of widely studied algebras, including Boolean algebras, Heyting algebras, MV-algebras, basic logic algebras and lattice-ordered groups. In the light of algebraization, the various structural rules correspond to analogous algebraic equations. For instance, the exchange rule for a logic corresponds to the commutativity of its algebraic models. One of the purposes of this thesis is to address the properties-of and relationships-between certain structural rules and their algebraic counterparts.

Within the substructural logic framework, Gentzen was able to prove the decidability of propositional intuitionistic logic $\mathbf{FL}_{\mathsf{ecw}}$ in the 1930s. It remained unknown whether any "natural" propositional logics were undecidable, outside of directly constructing logics for this purpose. A first surprising breakthrough comes when Urquhart showed that the propositional relevance logic $\mathbf{R}$ was undecidable in the late 1980s. However, $\mathbf{R}$ is not an extension of $\mathbf{FL}$ by structural rules due to the fact that $\mathbf{R}$ is distributive. A major breakthrough within this framework came when $\mathbf{FL}_{\mathsf{c}}$ was shown to be undecidable by Chvalovský and Horčík in 2016. In contrast, the question of whether any structural extensions of $\mathbf{FL}_{\mathsf{e}}$ are undecidable has remained an open problem. Actually, $\mathbf{FL}_{\mathsf{e}}$ and many of its structural extensions were shown to be decidable. The main results of this thesis resolves this problem by demonstrating the undecidability for an infinite class of such logics.

Approaches for proving decidability come in many different flavors, whether it be syntactical versus semantical analysis, or a constructive versus nonconstructive argument. In this thesis we utilize all such techniques. In the presence (or absence) of specific structural

rules in each case, we provide constructive syntactic proofs for decidability and nonconstructive algebraic proofs of decidability, as well as complexity upper bounds or lower bounds for such procedures.

In contrast to the variety of techniques for establishing decidability, proving undecidability almost always traces down to the same approach: encode some halting problem for Turing machines within the structure. However the difficulty is twofold. One must provide a suitable encoding of the machine as well as demonstrate that such an encoding is faithful. In this thesis, we present a general theory for encoding decision problems in residuated structures. From the substructural logic perspective, in the presence (or absence) of specific structural rules we prove that deducibility in that logic is undecidable. In particular, we demonstrate that provability is undecidable establishing the claim mentioned above. In this way, we demonstrate the undecidability for an infinitude of nonclassical propositional logics.

1.1 Chapter summaries

This chapter serves as both the theoretical and historical context for this thesis. In the preliminaries section we develop the formal background for the objects of study. In particular, we recall basic definitions and propositions about ordered algebraic structures and substructural logics. Specifically the variety of (commutative) residuated lattices $(\mathsf{C})\mathsf{RL}$, the Full Lambek calculus $\mathbf{FL}$, and the intimate connections of these two structures via algebraization. Particularly, the syntactic notions of provability and deducibility are semantically rendered as satisfaction for the equational and quasi-equational theories, respectively, for varieties of residuated lattices. Most of this background can be found in the standard monograph [9].

Chapter 2 concerns properties of equations in the $\{\vee, \cdot, 1\}$-fragment of residuated lattices, as well as their relation to structural rules for substructural logics.[1] It is here that the theory of *residuated frames* is first introduced, as developed by Galatos and Jipsen in [8], for it will serve as an essential technical tool for the entirety of this paper. In particular, we will highlight the preservation of *simple equations* and their structural counterparts *simple rules* within residuated frames constructions. In Section 2.1, we present key definitions and propositions about equations in the signature $\{\vee, \cdot, 1\}$, which we call *basic idempotent semiring* (ISR)*-equations*, in the setting of both residuated lattices and idempotent semirings. It is here where simple equations and simple structural rules find their definition. In Section 2.2, we recall residuated frames and their preservation of simple equations as seen in [8]. In Section 2.3, we investigate when simple equations are consequences of others. Through a straightforward residuated frames construction, we achieve Theorem 2.3.4 in particular, which essentially states that the $\{\vee, \cdot, 1\}$-fragment of the equational theory for the variety $\mathsf{RL} + \Sigma$ coincides with the equational theory of $\mathsf{ISR} + \Sigma$, where Σ is a set of simple equations. This construction also provides a recursively enumerable procedure for determining whether one equation implies another, often called the *subvariety containment problem*. In Section 2.4, we inspect some widely-studied classes of simple equations. Using the results from the previous section, we demonstrate some characterizations that will be useful for the remaining chapters e.g., Theorem 2.4.1 and Corollary 2.4.3. Lastly, in Section 2.5 we prove a deduction theorem for so-called *expansive* varieties of commutative residuated lattices.[2] Corollary 2.5.2 will be needed for the remaining chapters, specifically for bootstrapping the undecidability of the quasi-equational theory to undecidability of the equational theory for such residuated lattices.

---

[1] The following footnotes of this section will contain examples of $\{\vee, \cdot, 1\}$-(in)equations. These are meant to be read as the variety $\mathcal{V} + (e)$, where (e) is such an equation and $\mathcal{V}$ some variety understood in context.

[2] E.g., $x \leq x^2$ or $x \leq x^2 \vee x^3$

Chapter 3 establishes the decidability of many structures extended by the equations and structural rules presented in Chapter 2. In Section 3.1, we recall the *finite embeddability property* (FEP) and *finite model property* (FMP) for algebraic varieties and its relation to the decidability of universal theories. We also show how a result of Blok and van Alten [3] establishes the failure of the FEP for a collection of special simple equations in Proposition 3.1.3.[3] In Section 3.2, we illustrate how Theorem 2.3.4 provides a decision procedure for the $\{\vee, \cdot, 1\}$-fragment of the equational theory for many varieties in RL. In Section 3.3, we remark about the applicability of [8] for proving the FMP, and in Theorem 3.2.2 we establish the FMP for varieties extended by so-called *completely linear equations*.[4] Lastly, in Section 3.4 we present a decision procedure for the substructural logic counterpart of so-called *potent*-varieties, which are varieties satisfying some equation $x^n = x^{n+m}$. This is a generalization of the proof due to Gentzen [11] showing the decidability of $\mathbf{FL}_{\mathsf{ecw}}$. Furthermore, in Theorem 3.4.6 we show that this decision procedure is at worst double-exponential with respect to the number of symbols present in the input. Although such a procedure is computationally expensive, it is nevertheless primitive recursive. In contrast, the procedure for $\mathbf{FL}_{\mathrm{ec}}$ was shown to be non-primitive recursive by Urquhart [23], and even more dramatically, $\mathbf{FL}_{\mathsf{c}}$ was shown to be undecidable by Chvalovský and Horčík [5].

Chapter 4 begins our investigation of complexity lower bounds for satisfaction in the equational and quasi-equational theories for varieties of residuated lattices. At its heart, the techniques of this chapter are inspired by those found in [17, 23, 8, 14, 5]. In Section 4.1, we develop a general definition of *algebraic machines*. These machines are meant to encode the computations of some abstract mathematical machine as order relations in the algebra. Due to the fixed structure of a given machine, this correspondence relates to

[3] E.g., $x \leq x^2 \vee 1$ or $xy \leq x^2 \vee y^2$.

[4] E.g., $xy \leq x \vee y$ or $xyz \leq xy \vee yz \vee zx \vee x \vee y \vee z$.

the complexity of the *word problem* for these algebraic structures. Inspired by [14], we use a residuated frames construction to prove the completeness of this result, while the soundness is easily achieved since residuated lattices have semiring reducts. Furthermore, we introduce a notion of *admissibility* of simple equations for such machines. We will view instances of a simple equation $[\mathrm{R}]$ as "glitches" within the computations, and admissibility being a certain resiliency to such glitches. In this way, our residuated frames construction allows us to produce an algebra satisfying the equation, i.e., $\mathbf{W}^+ \in \mathsf{RL} + [\mathrm{R}]$, to serve as our countermodel for completeness. In Section 4.2 we introduce counter machines and their algebraic renderings. Since counter machines have an undecidable halting problem, we show that such a presentation proves the undecidability of the word problem for RL, particularly in its $\{\leq, \cdot, 1\}$-fragment. We also show that this same encoding establishes that certain weakenings of commutativity are admissible.[5] In Section 4.3 we present the algebraic rendering of *And-branching counter machines*, as invented in [17] to prove the undecidability of linear logic. At the cost of adding $\vee$ to the signature, this presentations allows for the construction of algebraic machines in which commutativity is admissible. As a consequence, this proves undecidability of the word problem, particularly for the $\{\vee, \cdot, 1\}$-fragment, for any variety $\mathcal{V}$ in the interval $\mathsf{CRL} \subseteq \mathcal{V} \subseteq \mathsf{RL}$. Lastly, in Section 4.4 we outline a construction due to Urquhart [23] establishing that any decision procedure for provability in $\mathbf{FL}_{\mathsf{ec}}$ cannot be primitive recursive. We show how this construction precisely fits within our framework of algebraic machines, and therefore naturally extends to a larger class of simple equations.

Chapter 5 is the demonstration of new undecidability results, utilizing the techniques developed in the previous chapter, for extensions of $(\mathsf{C})\mathsf{RL}$ by simple equations from a class $\mathcal{U}$. In Section 5.1, we provide a construction that can guarantee admissibility for any finite

[5] E.g., $x^2y^2 = y^2x^2$ or generally $x^ny^m = y^mx^n$ for any $n, m \geq 2$.

set of simple equations from $\mathcal{U}$. The main idea is essentially that, when viewed as glitches in a machine, members of $\mathcal{U}$ are well-behaved-enough in their effect on computations of a machine. That is, given any machine M and equation $[\mathrm{D}] \in \mathcal{U}$,[6] we can faithfully simulate the acceptance of M in a straightforward way by another machine M' so that $[\mathrm{D}]$ is admissible in M'. In this way, Corollary 4.1.10 guarantees the undecidability of the $\{\vee, \cdot, 1\}$-fragment of the word problem for $(\mathsf{C})\mathsf{RL} + [\mathrm{D}]$. This will prove Theorem 5.3.1, which also simultaneously demonstrates the undecidability of deducibility for the corresponding substructural logic $\mathbf{FL}_{\mathsf{e}} + (\mathrm{D})$. Consequently, using the deduction theorem from Section 2.5, our capstone Theorem 5.1.13 proves that the equational theory for $\mathsf{CRL} + [\mathrm{D}]$ is undecidable for any expansive $[\mathrm{D}] \in \mathcal{U}$. Equivalently, this shows that provability in the corresponding substructural logic $\mathbf{FL}_{\mathsf{e}} + (\mathrm{D})$ is undecidable. E.g., the equation $[\mathrm{D}] : x \leq x^2 \vee x^3$ is an expansive member of $\mathcal{U}$, so the equational theory of $\mathsf{CRL} + [\mathrm{D}]$ is undecidable, and therefore provability is undecidable in $\mathsf{FL}_{\mathsf{e}} + (\mathrm{D})$ where (D) is the structural rule

$$\frac{\Delta_1, \Gamma, \Gamma, \Delta_2 \Rightarrow \Pi \quad \Delta_1, \Gamma, \Gamma, \Gamma, \Delta_2 \Rightarrow \Pi}{\Delta_1, \Gamma, \Delta_2 \Rightarrow \Pi}\ (\mathrm{D}).$$

Section 5.2 proceeds in a similar way to Section 5.1, and aims at proving undecidability for the smaller ordered-monoid fragment of the word problem for RL. We show that this can be achieved, at least in general, for a class of equations $\mathcal{U}_{-1} \subseteq \mathcal{U}$ in Theorem 5.2.6.

Lastly, in Section 5.3 we provide a characterization for the class of equations $\mathcal{U}$ which is essential for both Theorem 5.3.1 and Theorem 5.2.6. The definition of $\mathcal{U}$ is equivalently stated via, $[\mathrm{D}] \in \mathcal{U}$ if and only if $\mathsf{CRL} + [\mathrm{D}] \not\models [V]$, for some *spinal equation* $[V]$ of the form:

$$[V] : x_1^{f(1)} \cdots x_k^{f(k)} \leq 1 \ \vee \ x_1^{v_1(1)} \ \vee \ x_1^{v_2(1)} x_2^{v_2(2)} \ \vee \cdots \vee \ x_1^{v_k(1)} \cdots x_k^{v_k(k)},$$

[6] E.g., $x \leq x^n \vee x^{n+m}$ for any $n, m > 0$.

for some $k \geq 1$ and vectors $f, v_1, \ldots, v_k \in \mathbb{N}^k$ such that $f \neq v_k$ and $v_i(i) > 0$ for each $i = 1, \ldots, k$. The goal of this section is to establish that such non-spinal equations satisfy a condition that guarantees admissibility for the machines defined in Sections 5.1 and 5.2. However, the techniques needed to prove this claim are quite distinct and unrelated to those needed in rest of the chapter, which is why they are presented last. We show that the property of satisfying a spinal equation is related to whether or not there exists positive solutions to some corresponding systems of linear equations in $\mathbb{R}^n$. Each joinand of an equation will be associated to some vector, and the right-hand side of simple equations as a set of vectors, which we may view as a matrix. In this context, monoid substitutions will also correspond to an associated matrix, and applications of a substitution as the transformation, or product, of this matrix with a vector (i.e., monoid term) or matrix (i.e., a finite join of monoid terms). In this way, a simple equation is a member of $\mathcal{U}$ if and only if its associated matrix does not appear in the decomposition of some spinal equation in terms of *upper-triangular block matrices*. Further, we show that this is equivalent to satisfying the sufficient condition of admissibility defined in Section 5.1.3.

Finally, in Chapter 6, we remark about the relationship of our results to related results known for non-commutative structures. We conclude by presenting a list of open problems for future research.

1.2 Preliminaries

By $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ we denote the set of *integers*, *rational numbers*, and *real numbers*, respectively. By $\mathbb{N}$ we denote the set of non-negative integers, i.e., *natural numbers*, by $\mathbb{Z}^+$ the set of positive integers. Let A, B, C be sets. The *powerset*, i.e., the set of all subsets of A, is denoted by $\wp(A)$. By $\mathrm{id}_A : A \to A$ we denote the identity map $a \mapsto a$ for all $a \in A$. We define B^A to be the set of all functions $f : A \to B$. If $f : A \to B$ and $g : B \to C$, their *composition* is written as $g \circ f : A \to C$, defined pointwise via $(g \circ f)(a) = g(f(a))$ for

each $a \in A$. For a function $f : A \to A$, we recursively define $f^0 := \mathrm{id}_A$ and $f^{n+1} := f \circ f^n$, for each $n \in \mathbb{N}$.

1.2.1 Ordered Sets. A structure $\mathbf{P} = (P, \leq_{\mathbf{P}})$ is a *preordered set*, or *preorder*, if $\leq_{\mathbf{P}}$ is a binary relation on Q such that, for all $x, y, z \in P$ the following hold:

- $x \leq_{\mathbf{P}} x$ (reflexivity),
- $x \leq_{\mathbf{P}} y$ and $y \leq_{\mathbf{P}} z$ imply $x \leq_{\mathbf{P}} z$ (transitivity).

$\mathbf{P}$ is called a *partially ordered set*, or *poset*, if $\mathbf{P}$ is a preorder which additionally satisfies the following for every $x, y \in P$:

- $x \leq_{\mathbf{P}} y$ and $y \leq_{\mathbf{P}} x$ imply $x = y$ (antisymmetry).

We will denote $\leq_{\mathbf{P}}$ simply by $\leq$ if it is understood unambiguously in context.

Let $\mathbf{P}$ and $\mathbf{Q}$ be posets. A map $f : P \to Q$ is said to be *monotone* if $x \leq_{\mathbf{P}} y$ implies $f(x) \leq_{\mathbf{Q}} f(y)$ for all $x, y \in P$. For $f : P \to P$, we say f is *expanding* if $x \leq f(x)$ for all $x \in P$, and *idempotent* if $f \circ f = f$. We call a map $\gamma : P \to P$ a *closure operator on* $\mathbf{P}$ if γ is expanding, monotone, and idempotent, and by $\mathbf{P}_\gamma$ we denote the poset of γ*-closed elements*, that is $P_\gamma := \gamma[P] = \{\gamma(p) : p \in P\}$.

A *Galois connection on* $\mathbf{P}$ *and* $\mathbf{Q}$ is a pair of maps $({}^{\triangleright}, {}^{\triangleleft})$, where ${}^{\triangleright} : P \to Q$ and ${}^{\triangleleft} : Q \to P$ such that $q \leq_{\mathbf{Q}} p^{\triangleright}$ iff $p \leq_{\mathbf{P}} q^{\triangleleft}$ for all $p \in P$ and $q \in Q$.

Proposition 1.2.1 ([9])**.** If $({}^{\triangleright}, {}^{\triangleleft})$ is a Galois connection on posets $\mathbf{P}$ and $\mathbf{Q}$, then the map $\gamma : P \to P$ defined by $\gamma(x) = x^{\triangleright\triangleleft}$ is a closure operator on $\mathbf{P}$.

Example 1.2.1. Given sets A, B and a relation $R \subseteq A \times B$, for sets $X \subseteq A$ and $Y \subseteq B$, we define

$$X \; R \, Y \iff x \; R \; y \text{ for all } x \in X \text{ and } y \in Y.$$

For $x \in A$ and $y \in B$, we write $x\ R\ Y$ and $X\ R\ y$ as abbreviations for $\{x\}\ R\ Y$ and $X\ R\ \{y\}$, respectively. Define ${}^{\triangleright} : \wp(A) \to \wp(B)$ and ${}^{\triangleleft} : \wp(B) \to \wp(A)$ via

$$X^{\triangleright} := \{y \in B : X\ R\ y\} \quad \text{and} \quad Y^{\triangleleft} := \{x \in A : x\ R\ Y\},$$

for all $X \in \wp(A)$ and $Y \in \wp(B)$. Then $({}^{\triangleright}, {}^{\triangleleft})$ forms a Galois connection on the posets $(\wp(A), \subseteq)$ and $(\wp(B), \subseteq)$, called the Galois connection *induced by* R.

1.2.2 Notions from Universal Algebra. Assuming familiarity with basic set-theoretical concepts, in this section we will recall the basic notions of Universal Algebra. We shall refer to [4] for a more detailed exposition.

Given a (non-empty) set A, a *n-ary operation* on A is any function f from A^n to A; the map $\sigma(f) = n$, that associates to a function symbol a natural number called the *arity* of f. The image of $(a_1, \ldots, a_n)$ under an n-ary operation f is denoted by $f(a_1, \ldots, a_n)$. An algebraic *type* is a pair $\mathscr{F} = (F, \sigma)$ of a set of function symbols F together with an arity map $\sigma : F \to \mathbb{N}$.

An *algebra* of type $\mathscr{F}$ is a pair $\mathbf{A} = (A, \langle f^{\mathbf{A}} \rangle_{f \in F})$ made of a domain set A and a family $\langle f^{\mathbf{A}} \rangle_{f \in F}$ of operations $f^{\mathbf{A}} : A^{\sigma(f)} \to A$. We will refer to them as the *fundamental* operations of $\mathbf{A}$. The underlying set A is often called the *universe* of the algebra. The superscripts of the operations will usually be omitted in the text, and we will often write the type of the algebra as the sequence $\langle \sigma(f^{\mathbf{A}}) \rangle_{f \in F}$.

By a *subalgebra* of $\mathbf{A}$ we mean an algebra $\mathbf{B} = (B, \langle f^{\mathbf{A}} \restriction_B \rangle_{f \in F})$ where $B \subseteq A$, where $f^{\mathbf{A}} \restriction_B$ is the restriction of $f^{\mathbf{A}}$ to B, and B is closed under the operations of $\mathbf{A}$, i.e. $f^{\mathbf{A}}(b_1, b_2, \ldots, b_{\sigma(f^{\mathbf{A}})}) \in B$, for all $b_1, \ldots, b_{\sigma(f^{\mathbf{A}})} \in B$. If $\mathscr{F}$ is a type and $G \subseteq F$, the *$\mathscr{G}$-reduct* of an algebra of type $\mathscr{F}$, $\mathbf{A} = (A, \langle f^{\mathbf{A}} \rangle_{f \in F})$, is the algebra $\mathbf{A}^{\mathscr{G}}$ with underlying set A and operations $\langle f^{\mathbf{A}} \rangle_{f \in G}$. A *partial algebra* $\mathbf{C}$ of $\mathbf{A}$ is any subset C of A equipped

with *partial* operations restricted to C, i.e., If $f^{\mathbf{A}}(a_1, \ldots, a_n) = c$ and $a_1, \ldots, a_n, b \in C$, then $f^{\mathbf{C}}(a_1, \ldots, a_n) = c$.

Suppose that $\mathbf{A}$ and $\mathbf{B}$ are two algebras of the same type $\mathscr{F}$. A mapping $h : A \to B$ is called a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ if for each f of arity n in F and every $a_1, \ldots, a_n \in A$,

$$h(f^{\mathbf{A}}(a_1, \ldots, a_n)) = f^{\mathbf{B}}(h(a_1), \ldots, h(a_n)).$$

If $\{\mathbf{A}_i : i \in I\}$ is a family of algebras of the same type, we define the *direct product* algebra $\prod_{i \in I} \mathbf{A}_i$, with universe the Cartesian product of the universes A_i, and fundamental operations defined by:

$$f^{\prod}(\langle a_{i1} \rangle_{i \in I}, \ldots, \langle a_{i\sigma(f)} \rangle_{i \in I}) = \langle f^{\mathbf{A_i}}(a_{i1}, \ldots, a_{i\sigma(f)}) \rangle_{i \in I},$$

for all $a_{ij} \in A_i$, $i \in I$ and $j \in \{1, \ldots, \sigma(f)\}$.

A class of algebras of the same type is called a *variety* if it is closed under homomorphic images, subalgebras and direct products. We shall refer to the variety generated by a class of algebras $\mathcal{K}$ as $\mathcal{V}(\mathcal{K})$. Let now $\mathcal{H}(\mathcal{K}), \mathcal{S}(\mathcal{K})$ and $\mathcal{P}(\mathcal{K})$ denote respectively the classes of homomorphic images, subalgebras and direct products of algebras in $\mathcal{K}$, then the following well known theorem due to Tarski holds.

Theorem 1.2.2 ([22])**.** For every class of algebras $\mathcal{K}$, $\mathcal{V}(\mathcal{K}) = \mathcal{HSP}(\mathcal{K})$.

Let X be a set of variables, $\mathscr{F}$ a type and $(X \cup F)^*$ the set of all finite sequences of elements of $X \cup F$. The set $T_{\mathscr{F}}(X)$ of terms in $\mathscr{F}$ over X is the least subset of $(X \cup F)^*$ that contains X and if $f \in F$ and $t_1, t_2, \ldots, t_{\sigma(f)} \in T_{\mathscr{F}}(X)$, then the sequence

$$f t_1 t_2 \ldots t_{\sigma(f)} \in T_{\mathscr{F}}(X).$$

The term algebra $\mathbf{T}_{\mathscr{F}}(X)$ is the algebra with underlying set $T_{\mathscr{F}}(X)$, type $\mathscr{F}$ and operations $f^{\mathbf{T}_{\mathscr{F}}(X)}$, for $f \in F$, defined by $f^{\mathbf{T}_{\mathscr{F}}(X)}(t_1, t_2, \ldots, t_{\sigma(f)}) = ft_1t_2 \ldots t_{\sigma(f)}$, for all $t_i \in T_{\mathscr{F}}(X)$.

If $\mathbf{A}$ is an algebra of type $\mathscr{F}$, t a term in $\mathscr{F}$ over a set of variables X and the variables occurring in t, denoted $\mathrm{supp}(t) := \{x_1, x_2, \ldots, x_n\}$, we define the *term operation* $t^{\mathbf{A}}$ of t inductively on the sub-terms of t to be the operation defined as follows: $x_i^{\mathbf{A}}$ is the i-th projection operation on A^n, and given $ft_1t_2 \ldots t_{\sigma(f)}$, where $f \in F$ and $t_1, t_2, \ldots, t_{\sigma(f)} \in T_{\mathscr{F}}(X)$, then $s^{\mathbf{A}}$ is defined by

$$s^{\mathbf{A}}(a_1, a_2, \ldots a_n) = f^{\mathbf{A}}\left(t_1^{\mathbf{A}}(a_1, a_2, \ldots a_n), t_2^{\mathbf{A}}(a_1, a_2, \ldots a_n), \ldots, t_{\sigma(f)}^{\mathbf{A}}(a_1, a_2, \ldots a_n)\right).$$

If $t_1, t_2, \ldots t_n$ are terms of $T_{\mathscr{F}}(X)$ and $n = |\mathrm{supp}(t)|$, then the *substitution* of $t_1, t_2, \ldots t_n$ into t is the element $t^{T_{\mathscr{F}}(X)}(t_1, t_2, \ldots t_n)$. If $\mathbf{A}$ is an algebra of type $\mathscr{F}$ and t a term in $\mathscr{F}$, then the operation $t^{\mathbf{A}}$ is called a *term operation*. Two algebras of possibly different types are called *term equivalent* if every operation of one is a term operation of the other.

An *equation*, or *identity*, of type $\mathscr{F}$ over a set of variables X is a pair of terms of $T_{\mathscr{F}}(X)$. If t, s are terms we write $t = s$ for the equation they define, instead of (t, s). We say that an equation $t = s$ in $\mathscr{F}$ over X is *valid* in an algebra $\mathbf{A}$ of type $\mathscr{F}$, or it is *satisfied by* $\mathbf{A}$, in symbols $\mathbf{A} \models t = s$, if $t^{\mathbf{A}} = s^{\mathbf{A}}$. The notion of validity is extended to classes of algebras and sets of equations. A set $\mathcal{E}$ of equations in a type $\mathscr{F}$ is said to be valid in, or satisfied by a class $\mathcal{K}$ of algebras of type $\mathscr{F}$, in symbols $\mathcal{K} \models \mathcal{E}$, if every equation of $\mathcal{E}$ is valid in every algebra of $\mathcal{K}$. Equations are preserved by subalgebras, homomorphic images and direct products. A theory of equations, or equational theory T in a type $\mathscr{F}$ is a congruence on $T_{\mathscr{F}}(X)$ closed under substitutions, i.e., if $(t = s) \in T, \mathrm{supp}(t) \cup \mathrm{supp}(s) = \{x_1, \ldots, x_n\}$, and $t_1, \ldots, t_n \in T_{\mathscr{F}}(X)$, then $(t^{T_{\mathscr{F}}(X)}(t_1, \ldots, t_n) = s^{T_{\mathscr{F}}(X)}(t_1, \ldots, t_n)) \in T$. It is easy to see that if $\mathcal{K}$ is a class of algebras of type $\mathscr{F}$, then $Th_{Eq}(\mathcal{K}) = \{(t = s) \in$

$T_{\mathscr{F}}(X) : \mathcal{K} \models t = s\}$ is an equational theory, called the *equational theory* of $\mathcal{K}$. Given a set $\mathcal{E}$ of equations of a similarity type $\mathscr{F}$ the equational class axiomatized by $\mathcal{E}$ is defined to be the class $Mod(\mathcal{E}) = \{A : A \models \mathcal{E}\}$ of algebras of type $\mathscr{F}$, that satisfy all equations of $\mathcal{E}$; the set $\mathcal{E}$ is called an *equational basis* for $Mod(\mathcal{E})$. By previous observations, every variety is an equational class. More precisely, the following well-known theorem due to Birkhoff holds.

Theorem 1.2.3 ([1])**.** For every class of algebras $\mathcal{K}$, $\mathcal{HSP}(\mathcal{K}) = Mod(Th_{Eq}(\mathcal{K}))$. Thus $\mathcal{K}$ is a variety iff it is the class of models of an equational theory.

1.2.3 Algebras and Varieties. Let A be a set. A function $* : A \times A \to A$ is called a *binary operation on* A, and we will write $a * b := *(a, b)$. We say $*$ is:

- *associative* iff $\forall a, b, c \in A$, $a * (b * c) = (a * b) * c$,
- *commutative* iff $\forall a, b \in A$, $a * b = b * c$, and
- *idempotent* iff $\forall a \in A$, $a * a = a$.

We say an element $1 \in A$ is an *identity element for* $*$ if for all $a \in A$, $a * 1 = 1 * a = a$. An algebra $\mathbf{S} = (S, *)$ is called a *semigroup* if $*$ is an associative binary operation on S. Note that if a semigroup has an identity then the identity is unique.[7] A structure $\mathbf{M} = (M, *, 1)$ is called a *monoid* if $(M, *)$ is a semigroup where 1 is the identity element for $*$. We say $\mathbf{S}$ ($\mathbf{M}$) is a commutative or idempotent semigroup (monoid) if $*$ is commutative or idempotent, respectively.

A commutative idempotent semigroup $\mathbf{S}$ is also known as a *semilattice*. The structure $\mathbf{S} = (S, \vee)$ is called a $\vee$-semilattice, where $\vee$ is called *join*. We often call the term $a \vee b$ the *least upper bound* of a and b, where we define the relation $\leq_\vee$ on S via $a \leq_\vee b$ iff $a \vee b = b$, for all $a, b \in S$. We see that $\leq_\vee$ is reflexive since $\vee$ is idempotent, it is antisymmetric since

[7] If e, e' are identities for $*$ then $e = e * e' = e'$.

$\vee$ is commutative, and transitive since $\vee$ is associative. Hence $(S, \leq)$ is a poset. Similarly, $(T, \wedge)$ is called a $\wedge$-semilattice, where $a \wedge b$ denotes the *greatest lower bound* of a and b, with the relation $\leq_\wedge$ on T via $a \leq_\wedge b$ iff $a \wedge b = a$, and deduce that $(T, \leq)$ is a poset.[8] If the $\vee$-semilattice [resp. $\wedge$-semilattice] is a monoid, we will represent the identity for $\vee$ [resp. $\wedge$] by the *falsum* symbol $\bot$ [resp. *verum* symbol $\top$], and call such a structure $(S, \vee, \bot)$ a $\bot$-bounded semilattice [resp. $(S, \wedge, \top)$ is $\top$-bounded].

Proposition 1.2.4. Let $\mathbf{S} = (S, \vee)$ be a $\vee$-semilattice. Then for all $a, b, c \in S$, *(i)* $a \leq a \vee b$ and *(ii)* $a \vee b \leq c$ implies $a \leq c$ and $b \leq c$. Similarly, if $(T, \wedge)$ is a $\wedge$-semilattice, then for all $a, b, c \in S$ *(iii)* $a \wedge b \leq a$ and *(iv)* $c \leq a \wedge b$ implies $c \leq a$ and $c \leq b$

Proof. (i) $a \vee b = (a \vee a) \vee b = a \vee (a \vee b)$. (ii) $a \leq a \vee b$ and $b \leq a \vee b$, and so by transitivity, $a \leq c$ and $b \leq c$. Both (iii) and (iv) follow by similar arguments. □

If $+$ and $\cdot$ are operations on a set A, we say A is *left [right]* $(\cdot, +)$*-distributive* if for all $a, b, c \in A$, $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ $[(b + c) \cdot a = (b \cdot a) + (c \cdot a)]$. We say A is $(\cdot, +)$-*distributive* if it is both left and right $(\cdot, +)$-distributive. Henceforth, when using symbol $\cdot$ we will write $ab := a \cdot b$, and will assume $\cdot$ binds more tightly than any other operation so to remove parenthesis and render expressions easier to read, e.g., $ab + cd := (a \cdot b) + (c \cdot d)$.

We call an algebra $\mathbf{R} = (R, +, \cdot, 1)$ a *semiring* if $(R, +)$ is a commutative semigroup, $(R, \cdot, 1)$ is a monoid, and R is $(\cdot, +)$-distributive. We say $\mathbf{R}$ is commutative if $(R, \cdot)$ is commutative, and idempotent if $(R, +)$ is idempotent. Semirings form a variety, and therefore so do (commutative) idempotent semirings. We denote the variety of (commutative) idempotent semirings by $(\mathsf{C})\mathsf{ISR}$. We call an algebra $\mathbf{R} = (R, +, \cdot, 0, 1)$ a *semiring with zero* if $(R, +, \cdot, 1)$ is a semiring where $(R, +, 0)$ is a monoid and $0x = x0 = 0$ for all $x \in R$.

[8] Note that the relations $\leq_\vee$ and $\leq_\wedge$ defined from the same operation on a commutative idempotent semigroup $\mathbf{S}$ are *dual*, i.e., $a \leq_\vee b$ iff $b \leq_\wedge a$

We will only be interested in the variety of idempotent semirings with zero, denoted $\mathsf{ISR}_\bot$, where we will use the falsum symbol $\bot$ to denote additive identity. [9]

A structure $\mathbf{G} = (G, \cdot, 1, \leq)$ a *partially-ordered monoid* if $(G, \cdot, 1)$ is a monoid and $(G, \leq)$ is a poset such that multiplication is order-preserving, i.e., $x \leq y$ implies $xz \leq yz$ and $zx \leq zy$ for all $x, y, z \in G$.

Proposition 1.2.5. If $(R, \vee, \cdot, 1)$ be an idempotent semiring, then multiplication is order preserving and hence $(R, \cdot, 1, \leq)$ is a partially-ordered monoid.

Proof. Let $x, y, z \in R$ and suppose $y \leq z$. By defintion, $z = y \vee z$, and so $xz = x(y \vee z) = xy \vee xz$. Hence $xy \leq xz$. Similarly, we deduce $yx \leq zx$. □

An algebra $\mathbf{L} = (L, \wedge, \vee)$ is a *lattice* if $(L, \wedge)$ and $(L, \vee)$ are $\wedge$ and $\vee$-semilattices, respectively, that satisfy the following absorption laws each $x, y \in L$:

- $x \vee (x \wedge y) = x$,
- $x \wedge (x \vee y) = x$.

We see that the $\wedge$ and $\vee$-semilattice orders coincide since $x \wedge y = x$ iff $y = x \vee y$. A lattice is $\bot$-*bounded* (resp. $\top$-*bounded*) if the $\vee$-semilattice [resp. $\wedge$-semilattice] reduct is, and a lattice is called *bounded* if it is both $\top$ and $\bot$-bounded. A lattice $\mathbf{L}$ is said to be *distributive* if it is both $(\vee, \wedge)$ and $(\wedge, \vee)$-distributive.[10]

[9] In the literature, a semiring is often defined to include a constant 0 in the signature such that $(R, +, 0)$ is a commutative monoid with $x0 = 0x = 0$ for all $x \in R$, and may or may not include the constant 1 in the signature, the latter only stipulating that $(R, \cdot)$ is a semigroup. For our purposes, we wish to include the multiplicative unit 1, while the inclusion of the additive unit 0 is unnecessary for the results that follow. However, we note that for the cases we consider, namely the $\{\vee, \cdot, 1\}$-reduct of residuated lattices, the existence of an additive unit $\bot$ will have the property $x\bot = \bot x = 0$.

[10] In fact, these conditions are equivalent in lattices. I.e., a lattice is distributive iff it is $(\vee, \wedge)$-distributive iff $(\wedge, \vee)$-distributive.

Definition 1.2.1. An algebra $\mathbf{R} = (R, \wedge, \vee, \cdot, \backslash, /, 1)$ is called a *(commutative) residuated lattice* if $(R, \wedge, \vee)$ is a lattice, $(R, \cdot, 1)$ is a (commutative) monoid and $(\backslash, /)$ is a pair of binary operations satisfying the following *law of residuation* for all $x, y, z \in R$:

$$xy \leq z \iff x \leq z/y \iff y \leq x\backslash z.$$

The *residual* operations $\backslash$ and $/$ are called, respectively, *left* and *right implication*, and can be viewed as a weaker notion of left and right division. In this way, we say a is the *numerator* and b the *denominator* in both the terms $b\backslash a$ and a/b. We prove the following proposition for the sake of completeness, however a more comprehensive exposition of such facts can be found in [8].

Proposition 1.2.6. Let $\mathbf{R} = (R, \wedge, \vee, \cdot, \backslash, /, 1)$ be a residuated lattice. The following hold:

1. For all $x, y \in R$, $x(x\backslash y) \leq y$ and $(x/y)y \leq x$.
2. Multiplication is order-preserving.
3. $\mathbf{R}$ is $(\cdot, \vee)$-distributive.
4. Implication is increasing in the numerator and decreasing in the denominator.
5. $\mathbf{R}$ is left $(\backslash, \wedge)$-distributive and right $(/, \wedge)$-distributive.
6. If $\mathbf{R}$ is commutative, then $x\backslash y = y/x$ for all $x, y \in R$.
7. If $\mathbf{R} \models 1 \leq x$ then $R = \{1\}$.

Proof. Let $x, y, z \in R$.

(1) $x(x\backslash y) \leq y$ iff $x\backslash y = x\backslash y$ by residuation, and so by symmetry $(x/y)y \leq x$.

(2) Suppose $x \leq y$. Since $yz \leq yz$, residuation entails $y \leq (yz)/z$. So $x \leq (yz)/z$ by transitivity with the assumption, and hence $xz \leq yz$ by residuation. By symmetry we obtain $zx \leq zy$.

(3) Since $y \leq y \vee z$ then $xy \leq x(y \vee z)$ by (2). Similarly, $xz \leq x(y \vee z)$, and hence $xy \vee xz \leq x(y \vee z)$. Let $c \in R$, then $x(y \vee z) \leq c$ iff $y \vee z \leq x \backslash c$ iff $y \leq x \backslash c$ and $z \leq x \backslash c$ iff $xy \leq c$ and $xz \leq c$ iff $xy \vee xz \leq c$. So by setting $c = xy \vee xz$ we deduce $x(y \vee z) = xy \vee xz$. By symmetry, we obtain $(y \vee z)x = yx \vee zx$ and so $\mathbf{R}$ is $(\cdot, \vee)$-distributive.

(4) Suppose $y \leq z$. By (1), $x(x \backslash y) \leq y \leq z$ by assumption, so $x \backslash y \leq x \backslash z$ by residuation. Similarly, using (2), $y \leq z$ implies $y(z \backslash x) \leq z(z \backslash x) \leq x$ by 1. Hence $z \backslash x \leq y \backslash x$ by residuation. By symmetry we obtain $y/x \leq z/x$ and $x/z \leq x/y$.

(5) By (4), division is increasing in the numerator implies $x \backslash (y \wedge z) \leq x \backslash y$ and $x \backslash (y \wedge z) \leq x \backslash z$, hence $x \backslash (y \vee z) \leq x \backslash y \wedge x \backslash z$. Fix $c \in R$, then by residuation, $c \leq x \backslash (y \wedge z)$ iff $xc \leq y \wedge z$ iff $xc \leq y$ and $xc \leq z$ iff $c \leq x \backslash y$ and $c \leq x \backslash z$ iff $c \leq x \backslash y \wedge x \backslash z$. Hence $x \backslash (y \wedge z) = x \backslash y \wedge x \backslash z$. Symmetrically, we obtain $(y \wedge z)/x = y/x \wedge z/x$.

(6) Suppose $\mathbf{R}$ is commutative. By (1), $x(x \backslash y) \leq y$, so by commutativity $(x \backslash y)x \leq y$ which implies $x \backslash y \leq y/x$. By symmetry, we obtain $x \backslash y = y/x$.

(7) Suppose for all $x \in R$, $1 \leq x$. Fix $x \in R$, then $1/x \in R$ so $1 \leq 1/x$ by assumption. By residuation, we obtain $x \leq 1$. Hence $x = 1$ and therefore $R = \{1\}$. □

When $\mathbf{R}$ is a commutative residuated lattice, we will write $x \to y := x \backslash y = y/x$. It is well known that (commutative) residuated lattices form a variety (see [9]), which we denote by $(\mathsf{C})\mathsf{RL}$. By Proposition 1.2.6, it is clear that the $\{\vee, \cdot, 1\}$-reduct of a (commutative) residuated lattice is a (commutative) idempotent semiring.

We say a residuated lattice is *distributive*, $\perp$*-bounded*, or *bounded* if the lattice reduct is distributive, $\perp$-bounded, or bounded, respectively, and denote these respective varieties by DRL, $\mathsf{RL}_{\perp}$, and BRL.

A *full Lambek algebra*, or *FL-algebra*, is a structure $\mathbf{A} = (A, \wedge, \vee, \cdot, \backslash, /, 1, 0)$ where $(A, \wedge, \vee, \cdot, \backslash, /, 1)$ is a residuated lattice and $0 \in A$ is some constant. Note that residuated lattices are exactly the 0-free reducts of FL-algebras. We denote the variety of FL-algebras

by FL, and commutative FL-algebras by $\mathsf{FL_e}$, where the naming convention will become apparent in the following section.

1.2.4 Inference rules and proofs. Let Q be a set. A subset $\vdash$ of $\wp(Q) \times Q$ is called a *consequence relation* over Q, if for every subset $X \cup Y \cup \{x, z\}$ of Q:

- if $x \in X$ then $X \vdash x$,
- if $X \vdash Y$ and $Y \vdash z$, then $X \vdash z$,

where $X \vdash x$ stands for $(X, x) \in \vdash$ and $X \vdash Y$ the proposition: $X \vdash y$ for all $y \in Y$. We note that for a consequence relation $\vdash$, the map $X \mapsto \{x \in Q : X \vdash x\}$ is a closure operator on $\wp(Q)$.

A *k-dimensional consequence relation over a structure* $\mathbf{S}$ is a consequence relation over S^k (we use the boldface to indicate there is additional structure). A consequence relation $\vdash$ on a structure $\mathbf{S}$ is *substitution invariant* if $X \vdash x$ implies $\boldsymbol{\sigma}(X) \vdash \boldsymbol{\sigma}(x)$, for every substitution $\boldsymbol{\sigma}$ on $\mathbf{S}$. For the purposes of this paper we only focus on (substitution invariant) k-dimensional consequence relations, where $k \in \{1, 2\}$.

A k-dimensional *inference rule over* $\mathbf{S}$ (or simply, a *rule*) is a pair $(r) = (t, T)$, where $T \cup \{t\}$ is a subset of S^k, and we write inference rules in fractional notation $\frac{T}{t}(r)$, where T is called the *premises* and t the *conclusion* of (r). If $T = \{t_1, \ldots, t_n\}$ we write

$$\frac{t_1 \quad \cdots \quad t_n}{t} \ (r),$$

where the premises are understood conjunctively. An *instance of a rule* (r) is obtained by applying a substitution $\boldsymbol{\sigma}$ to each term appearing in the rule, denoted by the pair $\frac{\boldsymbol{\sigma}[T]}{\boldsymbol{\sigma}(t)}(\boldsymbol{\sigma}, r)$.

A *proof* of s (*conclusion*) from (the set of) *assumptions* S in a set of rules R is a finite rooted tree with labeled vertices, defined inductively as follows:

1. Every element of S is a proof with that element as assumption and conclusion.

2. If $\boldsymbol{\sigma}$ is a substitution,

$$\frac{s_1 \quad \cdots \quad s_n}{s}\ (r)$$

is a rule in R, and $\mathsf{P}_1, \ldots, \mathsf{P}_n$ are proofs with conclusions $\boldsymbol{\sigma}(s_1), \ldots, \boldsymbol{\sigma}(s_n)$ and sets of assumptions $S_1, \ldots, S_n$, respectively, then

$$\frac{\mathsf{P}_1 \quad \cdots \quad \mathsf{P}_n}{\boldsymbol{\sigma}(s)}\ (\boldsymbol{\sigma}, r)$$

is a proof with a set of assumptions $S_1 \cup \cdots \cup S_n$ with conclusion s.

In this way, we see that the relation $\vdash_R$ over $\mathbf{S}$ defined by $X \vdash_R s$ iff there is a proof of s from assumptions X over the rules R, is a substitution invariant consequence relation.

1.2.5 The Full Lambek Calculus. We now recall the sequent system $\mathbf{FL}$, the *Full Lambek calculus*, which will serve as our basis for substructural logics. The *formulas* of $\mathbf{FL}$ are built from propositional variables $p, q, r, \ldots$ and constants 1 (unit) and 0 by using binary logical connectives $\cdot$ (fusion), $\backslash$ (right implication), $/$ (left implication), $\wedge$ (conjunction), and $\vee$ (disjunction). The set Fm of formulas is the smallest set containing the propositional variables and constants $0, 1$, and $(a * b) \in Fm$ for each $a, b \in Fm$ and connective $* \in \{\cdot, \backslash, /, \wedge, \vee\}$. $\mathbf{FL}$ *sequents* are expressions of the form $a_1, \ldots, a_m \Rightarrow b_1, \ldots, b_n$, where $m \geq 0$ and $1 \geq n \geq 0$. The rules of $\mathbf{FL}$ are displayed in Figure 1.1.

$$\frac{\Gamma \Rightarrow \alpha \quad \Delta_1, \alpha, \Delta_2 \Rightarrow \Pi}{\Delta_1, \Gamma, \Delta_2 \Rightarrow \Pi}\ (\mathbf{cut}) \qquad \frac{}{\alpha \Rightarrow \alpha}\ (\mathbf{init}) \qquad \frac{}{\Rightarrow 1}\ (1\mathrm{r})$$

$$\frac{\Gamma_1, \alpha, \beta, \Gamma_2 \Rightarrow \Pi}{\Gamma_1, \alpha \cdot \beta, \Gamma_2 \Rightarrow \Pi}\ (\cdot l) \qquad \frac{\Gamma \Rightarrow \alpha \quad \Delta \Rightarrow \beta}{\Gamma, \Delta \Rightarrow \alpha \cdot \beta}\ (\cdot r) \qquad \frac{\Gamma_1, \Gamma_2 \Rightarrow \Pi}{\Gamma_1, 1, \Gamma_2 \Rightarrow \Pi}\ (1l)$$

$$\frac{\Gamma \Rightarrow \alpha \quad \Delta_1, \beta, \Delta_2 \Rightarrow \Pi}{\Delta_1, \Gamma, \alpha \backslash \beta, \Delta_2 \Rightarrow \Pi}\ (\backslash l) \qquad \frac{\Gamma, \alpha \Rightarrow \beta}{\Gamma \Rightarrow \alpha \backslash \beta}\ (\backslash r) \qquad \frac{\Gamma \Rightarrow}{\Gamma \Rightarrow 0}\ (0r)$$

$$\frac{\Gamma \Rightarrow \alpha \quad \Delta_1, \beta, \Delta_2 \Rightarrow \Pi}{\Delta_1, \beta / \alpha, \Gamma, \Delta_2 \Rightarrow \Pi}\ (/l) \qquad \frac{\Gamma, \alpha \Rightarrow \beta}{\Gamma \Rightarrow \beta / \alpha}\ (/r) \qquad \frac{}{0 \Rightarrow}\ (0l)$$

$$\frac{\Gamma_1, \alpha, \Gamma_2 \Rightarrow \Pi \quad \Gamma_1, \beta, \Gamma_2 \Rightarrow \Pi}{\Gamma_1, \alpha \vee \beta, \Gamma_2 \Rightarrow \Pi}\ (\vee l) \qquad \frac{\Gamma \Rightarrow \beta}{\Gamma \Rightarrow \alpha \vee \beta}\ (\vee r_1) \qquad \frac{\Gamma \Rightarrow \alpha}{\Gamma \Rightarrow \alpha \vee \beta}\ (\vee r_2)$$

$$\frac{\Gamma_1, \beta, \Gamma_2 \Rightarrow \Pi}{\Gamma_1, \alpha \wedge \beta, \Gamma_2 \Rightarrow \Pi}\ (\wedge l_1) \qquad \frac{\Gamma_1, \alpha, \Gamma_2 \Rightarrow \Pi}{\Gamma_1, \alpha \wedge \beta, \Gamma_2 \Rightarrow \Pi}\ (\wedge l_2) \qquad \frac{\Gamma \Rightarrow \alpha \quad \Gamma \Rightarrow \beta}{\Gamma \Rightarrow \alpha \wedge \beta}\ (\wedge r)$$

Figure 1.1: Inference rules of $\mathbf{FL}$

The inference rules are presented in terms of *meta-variables*, where the letters α, β stand for formulas and are called *meta-formulas*, $\Gamma, \Delta, \ldots$ stand for finite (possibly empty) sequences of formulas called *meta-sequences*, and Π stands for either a formula or the empty-sequence, and is called a *stoup*. A *meta-sequent* $\mathbf{s}$ is given by $\Upsilon \Rightarrow \Psi$, where Υ is a specific sequence of meta-variables and Ψ is either empty, a meta-variable for formulas or sequences of sequences of formulas. An *assignment* ν is a substitution from meta-variables to sequences of formulas (separated by commas) of the appropriate type. If $\mathbf{s}$ is the meta-sequent $\Upsilon \Rightarrow \Psi$, then $\nu(\mathbf{s})$ is the sequent $\nu(\Upsilon) \Rightarrow \nu(\Psi)$.

In this way, proofs in $\mathbf{FL}$ are defined as above and $\vdash^{seq}_{\mathbf{FL}}$ is a (2-dimensional) substitution invariant consequence relation over Fm. If $\Phi \cup \{\psi\}$ is a set of formulas, we write $\Phi \vdash_{\mathbf{FL}} \psi$ if $\{ \Rightarrow \phi : \phi \in \Phi\} \vdash^{seq}_{\mathbf{FL}} \Rightarrow \psi$.

A *structural rule* is any rule (R) of the form for $n \geq 0$:

$$\frac{\Upsilon_1 \Rightarrow \Psi_1 \quad \cdots \quad \Upsilon_n \Rightarrow \Psi_n}{\Upsilon_0 \Rightarrow \Psi_0}\ (R),$$

where each Υ_i is a specific sequence of meta-variables and each Ψ_i is either empty, a meta-variable, or sequence of meta-variables (see [6]).

A *substructural logic* $\mathbf{L}$ over $\mathbf{FL}$ is an axiomatic extension of $\mathbf{FL}$ (by some set of axiom schemes). The extensions we primarily consider in the paper are those by sets of special structural rules called *simple rules*.[11] We write $\vdash_{\mathbf{L}}$ to denote the substitution invariant consequence relation defined by $\mathbf{L}$ in the usual way. We will be primarily interested in those extension of $\mathbf{FL}$ by some set Σ of structural rules, denoted by $\mathbf{FL}_{\Sigma}$. A few examples of widely studied structural rules are:

$$\frac{\Gamma, \alpha, \beta, \Delta \Rightarrow \Pi}{\Gamma, \beta, \alpha, \Delta \Rightarrow \Pi}\ (\mathsf{e}) \quad \frac{\Gamma, \Delta \Rightarrow \Pi}{\Gamma, \alpha, \Delta \Rightarrow \Pi}\ (\mathsf{w}) \quad \frac{\Gamma, \alpha, \alpha, \Delta \Rightarrow \Pi}{\Gamma, \alpha, \Delta \Rightarrow \Pi}\ (\mathsf{c}),$$

where (e) is called *exchange*, (w) *weakening*, and (c) *contraction*. The structure $\mathbf{FL_{ecw}}$ is the Gentzen calculus for intuitionistic logic, commonly denoted by $\mathbf{LJ}$. Algebraically, (e) corresponds to commutativity ($xy = yx$), (w) to integrality ($x \leq 1$), and (c) to square-increasing *contraction* ($x \leq x^2$).

All three relations $\vdash^{seq}_{\mathbf{FL}}$, $\vdash_{\mathbf{FL}}$, and $\models_{\mathsf{FL}}$ are equivalent (see [9],[10]), and this fact is known as the *algebraization* of $\mathbf{FL}$, in the sense of Blok and Pigozzi [2]. The translation between sequents, formulas, and equations can be given as follows: For a given sequent $\alpha_1, \ldots, \alpha_n \Rightarrow \alpha$, the corresponding equation and formula are $\alpha_1 \cdots \alpha_n \leq \alpha$ and $(\alpha_1 \cdots \alpha_n) \backslash \alpha$; for $\alpha_1, \ldots, \alpha_n \Rightarrow$ we put $\alpha_1 \cdots \alpha_n \leq 0$ and $(\alpha_1 \cdots \alpha_n) \backslash 0$. To a formula α, we associate $\Rightarrow \alpha$ and $1 \leq \alpha$. And to an equation $s = t$ we identify the formula $s \backslash t \wedge t \backslash s$

[11] See Section 2.1.1.

and the sequent $\Rightarrow s\backslash t \wedge t\backslash s$ (by $s \leq t$ we associate $s\backslash t$ and the sequent $s \Rightarrow t$). In light of this algebraization, we have that for a set of sequents $S \cup \{s\}$,

$$S \vdash^{seq}_{\mathbf{FL}} s \quad \text{iff} \quad \epsilon[S] \models_{\mathsf{FL}} \epsilon(s),$$

where $\epsilon(s)$ is the equation corresponding to s, and for every set of equations $E \cup \{\epsilon\}$,

$$E \models_{\mathsf{FL}} \epsilon \quad \text{iff} \quad s[E] \vdash^{seq}_{\mathbf{FL}} s(\epsilon).$$

where $s(\epsilon)$ is the sequent corresponding to ϵ.

If $\mathbf{L}$ is a substructural logic, by $\mathbf{L}^+$ we denote the 0-*free fragment* of $\mathbf{L}$. The equivalent algebraic semantics of $\mathbf{FL}^+$ are given by RL.

1.2.6 Notions of Decidability.

A substructural logic $\mathcal{L}$ has a decidable deducibility relation if there is an algorithm that decides whether $\Phi \vdash_{\mathcal{L}} \{\psi\}$, for all sets $\Phi \cup \{\psi\}$ of formulas. A class of algebras has a decidable (quasi)equational theory if there is an algorithm that decides whether a (quasi)equation holds in the class or not. Note that decidability problems for varieties of FL-algebras axiomatized by 0-free sets of equations reduce to the corresponding problems for the varieties of residuated lattices axiomatized by the same equations.

Chapter 2: Equations in the signature $\{\vee, \cdot, 1\}$

In this chapter, we will examine properties of equations in the $\{\vee, \cdot, 1\}$-fragment of residuated lattices. It is here that the theory of *residuated frames* [8] is first introduced. In particular, we highlight the preservation of *simple equations* and their structural counterparts *simple rules* within residuated frames constructions, which will be essential for the following chapters. In the first section, we present key definitions and propositions about equations in the signature $\{\vee, \cdot, 1\}$ in the setting of both residuated lattices and idempotent semirings. It is here where simple equations and simple structural rules find their definition. In the second section, we recall residuated frames and their preservation of simple equations. The third section investigates when simple equations are consequences of others, often called the *subvariety containment problem*. In particular, we exhibit a recursively enumerable procedure for determining whether one equation implies another. We achieve Theorem 2.3.4 in particular, which essentially states that the $\{\vee, \cdot, 1\}$-fragment of the equational theory for the variety $\mathsf{RL} + \Sigma$ coincides with the equational theory of $\mathsf{ISR} + \Sigma$, where Σ is a set of simple equations. The fourth section inspects some widely-studied classes of simple equations, in particular so-called knotted equations. Using the results from the previous section, we demonstrate some characterizations that will be useful for the remaining chapters e.g., Theorem 2.4.1 and Corollary 2.4.3. In the last section, we prove a deduction theorem for so-called *expansive* varieties of commutative residuated lattices. In particular, Corollary 2.5.2 will be needed for the remaining chapters, specifically for bootstrapping the undecidability of the quasi-equational theory to undecidability of the equational theory for such residuated lattices.

2.1 Equations in ISR

Since RL has a semiring reduct, an equation over $\{\vee, \cdot, 1\}$ is equivalent to an equality between two finite joins of monoid terms by distributivity. Since RL also has a $\vee$-semilattice reduct, by Proposition 1.2.4, such an equality is ISR-equivalent to a conjunction of inequations, which we call ISR-*equations*. In the following sections, we will let Var be a countable set of variables. For a subset $X \subseteq \mathsf{Var}$ we will denote by $X^* := T_{\{\cdot,1\}}(X)$ the set of monoid terms generating by X, and by $X^{*\vee}$ the free semiring generate by X. Since $\mathsf{Var}^{*\vee}$ is a semiring structure, every element of $\mathsf{Var}^{*\vee}$ can be written as a join of monoid words over Var^*.

Given a term $t \in \mathsf{Var}^*$, we define the *support of* t to be the set $\mathrm{supp}(t) \subseteq \mathsf{Var}$ containing exactly those distinct variables which occur in t, i.e., $t \in \mathrm{supp}(t)^*$ but $t \notin Y^*$ for any $Y \subsetneq \mathrm{supp}(t)$. By definition, $t \in \mathsf{Var}^*$ implies $\mathrm{supp}(t)$ is finite. For $Y \subseteq \mathsf{Var}^*$, let $\mathrm{supp}(Y)$ be the set of exactly those distinct variables which occur in elements of Y, i.e. $\mathrm{supp}(Y) = \bigcup_{t \in Y} \mathrm{supp}(t)$. Similarly, for terms $t_1, \ldots, t_n \in \mathsf{Var}^*$, by $\mathrm{supp}(t_1 \vee \cdots t_n)$ set of exactly those distinct variables which occur in each joinand of t_i, i.e., $\mathrm{supp}(t_1 \vee \cdots t_n) = \mathrm{supp}(\{t_1, \ldots, t_n\})$.

Definition 2.1.1. Let $m \geq 1$, $t_0, \ldots, t_m \in \mathsf{Var}^*$ be monoid terms. A universally quantified inequation $[A]$ of the form $t_0 \leq t_1 \vee \cdots \vee t_m$ is called a ISR-*equation*, and in this way we write $[A] = (t_0, A)$, where $A = \{t_1, \ldots, t_m\}$. A ISR-equation $[A] = (t_0, A)$ is called:

- *trivial* if $t_0 \in A$,
- *linear* if t_0 is linear, i.e., each variable appearing in t_0 occurs exactly once.[1],
- *proper* if $A \subseteq \mathrm{supp}(t_0)^*$,
- *integral* if $\mathrm{supp}(t_0) \setminus \mathrm{supp}(A)$ is nonempty,

[1]This can be stated via $t_0 = \prod_{x \in \mathrm{supp}(t_0)} x$, since Π is well-defined by commutativity.

- *degenerate* if $t \notin \mathrm{supp}(t_0)^*$ for each $t \in A$, namely every $t \in A$ contains a variable not appearing in t_0, and
- a *simple equation* if $[A]$ is a proper linear ISR-equation.[2]

If $\boldsymbol{\sigma}$ is a substitution, then $[\boldsymbol{\sigma}A] := (\boldsymbol{\sigma}(t), \boldsymbol{\sigma}[A])$, i.e., $[\boldsymbol{\sigma}A] : \boldsymbol{\sigma}(t_0) \leq \boldsymbol{\sigma}(t_1 \vee \cdots \vee t_m)$.

Since RL has an ISR-reduct, the following is immediate:

Proposition 2.1.1. Let $\Gamma \cup \{[A]\}$ be a set of ISR-equations. Then $\mathsf{ISR} + \Gamma \models [A]$ implies $\mathsf{RL} + \Gamma \models [A]$.

When understood in context, we will refer to an ISR-equation simply as an *equation*. Through a process called *linearization*, as shown in [8], we can prove an equation is equivalent to a linear equation:

Proposition 2.1.2. The following hold:

1. In ISR, every finite set of ISR-equations is equivalent to an ISR-equation.
2. In ISR, every ISR-equation is equivalent to a linear equation.
3. In ISR, every integral equation entails integrality ($x \leq 1$).
4. In RL and $\mathsf{ISR}_\perp$, every degenerate equation is equivalent to $1 \leq x$.
5. In RL and $\mathsf{ISR}_\perp$, every non-degenerate equation is equivalent to a simple equation.

Proof. (1) Let $t_i \leq u_i$, for $i = 1, \ldots, n$ be ISR-equations where $t_i \in \mathsf{Var}^*$ and $u_i \in \mathsf{Var}^{*\vee}$. By choosing fresh variables, we can assume that their sets of variables are disjoint. We claim the set $\{t_i \leq u_i : 1 \leq i \leq n\}$ is equivalent to the equation $t_1 \cdots t_n \leq u_1 \cdots u_n$, in which case we can even distribute on the right-hand side. The forward direction is obtained since multiplication is order preserving. The converse is obtained by, for each $1 \leq i \leq n$,

[2]Note that if $[A]$ is a simple equation and $t_0 = 1$ then $A = \{1\}$.

substituting 1 for each variable not appearing in t_i, u_i. This substitution yields exactly $t_i \leq u_i$, since the variables was assumed to be distinct.

(2) Fix an equation $[A]$ given by $t \leq u$, for some monoid term $t \in \mathsf{Var}^*$ and $u \in \mathsf{Var}^{*\vee}$. For each variable x appearing in t, we consider fresh variables $x_1, \ldots, x_n$ not appearing in t, u. Substitute $x_1 \vee \cdots \vee x_n$ for x in $[A]$ and distribute on both sides of $\leq$. So, if $t = vx^n w$, we obtain

$$vx^n w \leq u \implies v(x_1 \vee \cdots \vee x_n)^n w \leq u' \implies vx_1 \cdots x_n w \leq u',$$

where u' is obtained by the substitution $x \mapsto x_1 \vee \cdots \vee x_n$ and the last implication hold by distribution and the fact that $a \vee b \leq c$ implies $a \leq c$. The reverse direction is obtained by setting $x = x_1 = \cdots = x_n$, producing $t \leq u$. Doing this for all variables in t produces a linear term.

(3) Suppose $t \leq u$ is integral. Then there exists a variable x in t that occurs nowhere in u. If $t_L \leq u_L$ is the linearization of $t \leq u$ as defined in (2), all the variables $x_1, \ldots, x_n$ appear only in t_L precisely once, and appear nowhere in u_L. Substitute all variables different from x_1 to 1. Then $x_1 \leq 1 \vee \cdots \vee 1$, which is equivalent to $x_1 \leq 1$ by the idempotency of $\vee$.

(4&5) By the method of linearization from (2), it is enough to consider the equation $[A] : s \leq t_1 \vee \cdots \vee t_n$ where s is linear. Let $J = \{t_1, \ldots, t_n\}$ and $J_d \subseteq J$ be the set of all joinands t_i such that t_i contains variables that do not appear in s. Note that $[A]$ is degenerate if and only if $J_d = J$. For $\mathsf{ISR}_\perp$, both (4) and (5) are obtained by the substitution $\boldsymbol{\sigma}$ mapping $x \mapsto \perp$ if $x \in \mathrm{supp}(J) \setminus \mathrm{supp}(s)$. This yields $\boldsymbol{\sigma}(t) = \perp$ for each $t \in J_d$, and $\boldsymbol{\sigma}(s) = s$. If $[A]$ is degenerate then $[A]$ implies $s \leq \perp$ (which trivially entails $1 \leq x$) and we are done. Otherwise, there are joinands t_i such that $\mathrm{supp}(t_i) \subseteq \mathrm{supp}(s)$, i.e., $J \setminus J_d \neq \emptyset$ implies

$\boldsymbol{\sigma}(t) = t$ for all $t \in J \setminus J_d$, and hence $[A]$ implies $s \leq \bigvee_{t \in J \setminus J_d} t$, a simple equation. We now proceed with the case of RL.

Now, if $[A]$ is degenerate, fix a fresh variable $x \in \mathsf{Var}$ and define the substitution $\boldsymbol{\tau}$ generated by $x \mapsto 1$, for each $x \in \mathrm{supp}(s)$, and $y \mapsto x \wedge 1$ for each $y \notin \mathrm{supp}(s)$. Then $\boldsymbol{\tau}(s) = 1$ and for each $t \in J$, and $\boldsymbol{\tau}(t) = (x \wedge 1)^{m_t}$ for some $m_t \geq 1$, since $J = J_d$. Hence $\boldsymbol{\tau}(t) \leq x$ since $(x \wedge 1)^{m_v} \leq x \wedge 1 \leq x$. Thus $1 = \boldsymbol{\tau}(s) \leq \bigvee_{t \in J} \boldsymbol{\tau}(t) \leq x$. Since $\mathsf{RL} + (1 \leq x)$ defines the trivial variety, it follows that $\mathsf{RL} \models [A]$ iff $\mathsf{RL} \models 1 \leq x$.

If $[A]$ is not degenerate then $J_d \subsetneq J$. Define $v = \bigvee_{t \in J \setminus J'} t$. For each $w \in J_d$, there exists $y_w \notin \mathrm{supp}(s)$ such that $w = u_w y_w v_w$, for some terms $u_w, v_w \in \mathsf{Var}^*$. Let $u'_w, v'_w \in \mathsf{Var}^*$ be terms obtained by replacing each $y \notin \mathrm{supp}(s)$ in u_w, v_w by 1, for each $w \in J_d$. Now $y \notin \mathrm{supp}(s)$, make the following substitution $\boldsymbol{\tau}$

$$y \mapsto 1 \wedge \bigwedge_{w \in J_d} u'_w \backslash (v / v'_w).$$

It follows that $\boldsymbol{\tau}(v) = v$ and for every $w \in J_d$, $\boldsymbol{\tau}(u_w) \leq u'_w$ and $\boldsymbol{\tau}(v_w) \leq v'_w$, since $\boldsymbol{\tau}(y) \leq 1$ for each $y \notin \mathrm{supp}(s)$. Furthermore, since $\boldsymbol{\tau}(y_w) \leq u'_w \backslash (\bar{v} / v'_w)$, we obtain $\boldsymbol{\tau}(w) \leq u'_w \boldsymbol{\tau}(y_w) v'_w \leq v$. In this way we obtain

$$s = \boldsymbol{\tau}(s) \leq^{[A]} \boldsymbol{\tau}(v) = \bigvee_{t \in J} \boldsymbol{\tau}(t) \leq \bigvee_{t \in J \setminus J_d} t.$$

Since $t \in \mathrm{supp}(s)^*$ for each $t \in J \setminus J_d$, we have that $[\boldsymbol{\tau} A]$ is a simple equation. Hence $\mathsf{RL} + [A] \models [\boldsymbol{\tau} A]$. The converse is obtained since $v \leq v \vee \bigvee_{w \in J_d} w$, and so $\mathsf{RL}+ \models [A]$. Therefore, $\mathsf{RL} + [\boldsymbol{\sigma} A] \models [A]$ iff $\mathsf{RL} + [A] \models [\boldsymbol{\sigma} A]$. □

For an indexing on $\mathsf{Var} = \{x_1, x_2, \ldots\}$, we define the *$n$-variable linear term* $\mathbf{1_n} \in \mathsf{Var}^*$ via $\mathbf{1_n} := \prod_{i=1}^{n} x_i$, and $\mathbf{1_0} := 1$. If $[A]$ is a simple equation, then $[A]$ is ISR-equivalent to some rule $[\mathrm{R}] = (\mathbf{1_n}, \mathrm{R})$ by simply indexing the set Var in a particular way. In this way,

when we represent ISR-equations by non-italicized letters we implicitly assume an indexing on Var, e.g., $[\mathrm{R}] = (\mathbf{1_n}, \mathrm{R})$ or $[\mathrm{A}] = (a_0, \mathrm{A})$. We define $\mathsf{Var}_\perp := \mathsf{Var} \cup \{\perp\}$, and $\mathsf{Var}^*_\perp$ to be the free monoid over $\mathsf{Var}_\perp$ where $\perp$ is an absorbing element, i.e., $\perp x = x \perp = \perp$ for all $x \in \mathsf{Var}^*_\perp$.

2.1.1 ISR-equations and structural rules. To each ISR-equation $[\mathrm{A}] = (a_0, \mathrm{A})$, we associate the following structural rule in $\mathbf{FL}$:

$$\frac{\{\Delta_1, a^{\mathbf{FL}}(\Gamma_1, \ldots, \Gamma_n), \Delta_2 \Rightarrow \Pi\}_{a \in \mathrm{A}}}{\Delta_1, a_0^{\mathbf{FL}}(\Gamma_1, \ldots, \Gamma_n), \Delta_2 \Rightarrow \Pi}\ (\mathrm{A}), \tag{2.1}$$

and vice versa, where $\mathrm{supp}(\mathrm{A} \cup \{a_0\}) = \{x_1, \ldots, x_n\}$. As described in Section 1.2.5, the relations $\vdash_{\mathbf{FL}+(\mathrm{A})}$ and $\models_{\mathsf{FL}+[\mathrm{A}]}$ are equivalent. We call a structural rule, as written above, a *simple rule* if $[\mathrm{A}]$ is a simple equation.

Proposition 2.1.3 ([8])**.** For any set of simple rules Σ, the cut rule is admissible in $\mathbf{FL} + \Sigma$.

2.2 Simple Equations and Residuated Frames

We recall the structures known as residuated frames, as developed in [8]. For our purposes, a *residuated frame* is a structure $\mathbf{W} = (W, W', N)$ where

- $(W, *, 1)$ is a monoid,
- W' is a set,
- $N \subseteq W \times W'$, often called the *Galois relation*, is *nuclear*, i.e., there exists $\backslash\!\backslash : W \times W' \to W'$ and $/\!/ : W' \times W \to W'$ such that for all $u, v \in W$ and $w \in W'$,

$$u * v \mathrel{N} w \quad \text{iff} \quad u \mathrel{N} w /\!/ v \quad \text{iff} \quad v \mathrel{N} u \backslash\!\backslash w.$$

The relation N defines a Galois connection $({}^{\triangleright}, {}^{\triangleleft})$ on $\wp(W), \wp(W')$, as defined in Example 1.2.1. Hence, $\gamma_N : \wp(W) \to \wp(W)$ defined by $\gamma(X) = X^{\triangleright\triangleleft}$ is a closure operator on

$\wp(W)$. In fact, a relation N is nuclear if and only if γ_N is a nucleus (see [8]), where a *nucleus* is a closure operator $\gamma : G \to G$ on a partially ordered groupoid $\mathbf{G}$ satisfying $\gamma(x)\gamma(y) \leq \gamma(xy)$ [or equivalently, $\gamma(\gamma(x)\gamma(y)) = \gamma(xy)$] for all $x, y \in G$.

Proposition 2.2.1 ([8])**.** Let $\mathbf{W}$ be a residuated frame. Then the structure

$$\mathbf{W}^+ := \big(\wp(W)_{\gamma_N}, \cap, \cup_{\gamma_N}, *_{\gamma_N}, \backslash, /, \gamma_N(\{1\})\big)$$

is a residuated lattice, where

$$X \cup_{\gamma_N} Y := \gamma_N(X \cup Y), \quad X \backslash Y := \{z \in W : X * \{z\} \subseteq Y\},$$
$$X *_{\gamma_N} Y := \gamma_N(X * Y), \quad Y / X := \{z \in W : \{z\} * X \subseteq Y\}.$$

and $X * Y := \{x * y \in W : x \in X, y \in Y\}$ for all $X, Y \in \wp(W)_{\gamma_N}$.

We note that $\mathbf{W}^+$ is a *complete*[3] residuated lattice. In fact, $\bot^{\mathbf{W}^+} := \gamma_N(\emptyset)$ is the least element in $\mathbf{W}^+$, and thus $X *_{\gamma_N} \bot^{\mathbf{W}^+} = \bot^{\mathbf{W}^+} *_{\gamma_N} X = \bot^{\mathbf{W}^+}$ for any $X \in W^+$.

Let $(W, *, 1)$ be a monoid, W' a set, and $N \subseteq W \times W'$. Define $\widetilde{W'} := W \times W' \times W$ and $\widetilde{N} \subseteq W \times \widetilde{W'}$ to be the relation given by

$$x \;\widetilde{N}\; (u, z, v) \quad \text{iff} \quad u * x * v \; N \; z,$$

for all $x, u, v \in W$ and $z \in W'$. We call (W, W', N) a *preframe* and $(W, \widetilde{W'}, \widetilde{N})$ the structure *induced* by the preframe (W, W', N).

Proposition 2.2.2. Let (W, W', N) be a preframe. Then $(W, \widetilde{W'}, \widetilde{N})$ is a residuated frame.

[3] A lattice $\mathbf{L}$ is complete if it is closed under arbitrary joins, written $\bigvee X \in L$ for every $X \subseteq L$. Equivalently, $\mathbf{L}$ is complete if it is closed under arbitrary meets, written $\bigwedge X \in L$ for every $X \subseteq L$.

Proof. Observe $\widetilde{N}$ is nuclear since $x*y\ \widetilde{N}\ (u,z,v)$ iff $x\ \widetilde{N}\ (u,z,y*v)$ iff $y\ \widetilde{N}\ (u*x,z,v)$, for all $x,y\in W$ and $(u,z,v)\in\widetilde{W'}$.[4] □

2.2.1 Preservation of simple equations. Let $\mathbf{W}=(W,W',N)$ be a residuated frame and $[\mathrm{A}]=(a_0,\mathrm{A})$ be an ISR-equation where $\mathrm{supp}(\mathrm{A}\cup\{a_0\})=\{x_1,\ldots,x_n\}$ for some $n\geq 0$. We write $\mathbf{W}\models(\mathrm{A})_{\mathbf{W}}$ if:

$$\frac{\{a^{\mathbf{W}}(u_1,\ldots,u_n)\ N\ v\}_{a\in\mathrm{A}}}{a_0^{\mathbf{W}}(u_1,\ldots,u_n)\ N\ v}\ (\mathrm{A})_{\mathbf{W}},$$

for all $u_1,\ldots,u_n\in W$ and $v\in W'$.

Proposition 2.2.3 ([8])**.** Let $[\mathrm{A}]$ be an ISR-equation and $\mathbf{W}$ a residuated frame. If $\mathbf{W}^+\models[\mathrm{A}]$ then $\mathbf{W}\models(\mathrm{A})_{\mathbf{W}}$.

Proposition 2.2.4 ([8])**.** If $[\mathrm{R}]$ is a simple equation and $\mathbf{W}$ a residuated frame, then $\mathbf{W}\models(\mathrm{R})_{\mathbf{W}}$ if and only if $\mathbf{W}^+\models[\mathrm{R}]$.

The above proposition is not true in general for non-proper linear ISR-equations.[5] To handle such equations, a sufficient condition on the frame $\mathbf{W}=(W,W',N)$ is that $\perp^{\mathbf{W}^+}\neq\emptyset$, i.e., the nuclear image of the emptyset is not the emptyset.

Lemma 2.2.5. Let $[\mathrm{A}]$ be a non-proper linear ISR-equation and $\mathbf{W}$ a residuated frame. If $\perp^{\mathbf{W}^+}\neq\emptyset$, then $\mathbf{W}\models(\mathrm{A})_{\mathbf{W}}$ implies $\mathbf{W}^+\models[\mathrm{A}]$.

Proof. Let $\mathbf{W}=(W,W',N)$, be a residuated frame, where $(W,*,1)$ is the monoid, such that $\perp^{\mathbf{W}^+}\neq\emptyset$, and suppose $\mathbf{W}\models[\mathrm{A}]$, for some non-proper linear $[\mathrm{A}]=(\mathbf{1_n},\mathrm{A})$ where

[4] That is $\backslash\backslash, /\!/$ are given by $x\backslash\backslash(u,z,v):=(u*x,z,v)$ and $(u,z,v)/\!/y:=(u,z,y*v)$.

[5] Proposition 2.2.3 can be found in Theorem 3.10 in [8]. However, we remark that Theorem 3.10 in [8] as stated is not true in general for residuated frames and special care must be taken for non-proper linear ISR-equations.

$\mathrm{supp}(\mathrm{A} \cup \{\mathbf{1_n}\}) = \{x_1, \ldots, x_k\}$ and $k > n$. Note that $u_\perp \in \perp^{\mathbf{W}^+}$ implies $u_\perp \, N \, v$ for all $v \in W'$. In fact, since $\perp^{\mathbf{W}^+}$ is absorbing in $\mathbf{W}^+$, $u_\perp \in \perp^{\mathbf{W}^+}$ implies $x * u_\perp * y \, N \, v$ for all $v \in W'$ and $x, y \in W$.

If $[\mathrm{A}]$ is degenerate, then $\mathrm{supp}(a) \setminus \mathrm{supp}(\mathbf{1_n}) \neq \emptyset$ for each $\in \mathrm{A}$. It is enough to show that $\mathbf{W}^+ \models 1 \leq x$. Since $\perp^{\mathbf{W}^+}$ is the least element, it suffices to show $\gamma_N(\{1\}) \subseteq \perp^{\mathbf{W}^+}$, or equivalently, $\emptyset^\triangleright \subseteq \{1\}^\triangleright$. Fix $u_\perp \in \perp^{\mathbf{W}^+}$, and set $\overline{u} = (u_i)_{i=1}^k$ where $u_i = 1$ for $i \leq n$ and $u_j = u_\perp$ for $n < j \leq k$. Then $a^{\mathbf{W}}(\overline{u}) = u^{k_a}$ for each $a \in \mathrm{A}$, where $k_a > 0$, and ${\mathbf{1_n}}^{\mathbf{W}}(\overline{u}) = 1$. Hence for every $v \in W'$, we obtain $a^{\mathbf{W}}(\overline{u}) \, N \, v$, and since $\mathbf{W} \models (\mathrm{A})_{\mathbf{W}}$, it follows that ${\mathbf{1_n}}^{\mathbf{W}}(\overline{u}) \, N \, v$, i.e., $1 \, N \, v$. Thus $\gamma_N(\{1\}) \subseteq \perp^{\mathbf{W}^+}$, hence $\mathbf{W}^+ \models 1 \leq x$.

Now, if $[\mathrm{A}]$ is non-degenerate, then $n > 0$ and the set $\mathrm{A}' := \{a' \in \mathrm{A} : \mathrm{supp}(a') \subseteq \mathrm{supp}(\mathbf{1_n})\}$ is nonempty. Then $[\mathrm{A}'] := (\mathbf{1_n}, \mathrm{A}')$ is a simple rule. We claim that $\mathbf{W} \models (\mathrm{A}')$. Let $u_1, \ldots, u_n \in W$ and $u_\perp \in \perp^{\mathbf{W}^+}$. Define $\overline{u} = (u_1, \ldots, u_n)$ and $\overline{u}' = (u'_i)_{i=1}^k$ where $u'_i = u_i$ for $i \leq n$ and $u_j = u_\perp$ for $n < j \leq k$. Note that ${\mathbf{1_n}}^{\mathbf{W}}(\overline{u}') = {\mathbf{1_n}}^{\mathbf{W}}(\overline{u})$ and ${a'}^{\mathbf{W}}(\overline{u}') = {a'}^{\mathbf{W}}(\overline{u})$ for each $a \in \mathrm{A}'$. Now, suppose for some $v \in W'$, ${a'}^{\mathbf{W}}(\overline{u}) \, N \, v$ for each $a' \in \mathrm{A}'$. Then $a^{\mathbf{W}}(\overline{u}') \, N \, v$ for every $a \in \mathrm{A}$ since $a \in \mathrm{A} \setminus \mathrm{A}'$ is such that $a^{\mathbf{W}}(\overline{u}') = x * u_\perp * y$, for some $x, y \in W$. Since $\mathbf{W} \models (\mathrm{A})_{\mathbf{W}}$, it follows that ${\mathbf{1_n}}^{\mathbf{W}}(\overline{u}') \, N \, v$. Hence $\mathbf{W} \models (\mathrm{A}')_{\mathbf{W}}$. Since $[A']$ is a simple equation, by Proposition 2.2.4, $\mathbf{W}^+ \models [\mathrm{A}']$. Since $\mathrm{A}' \subseteq \mathrm{A}$ and $\vee$ is increasing, it follows that $\mathbf{W}^+ \models [A]$. □

2.3 Subvariety Containment

We will now address the question: for given sets Σ, Σ' of simple equations, when does $\mathsf{RL} + \Sigma \models \Sigma'$, i.e., is $\mathsf{RL} + \Sigma \subseteq \mathsf{RL} + \Sigma'$? We will show this is equivalent to whether or not $\mathsf{ISR} + \Sigma \models \Sigma'$.

Let $[A] = (a_0, A)$ be an ISR-equation. We define the 1-dimensional inference rule $(A)_{\mathsf{Var}^*}$ on Var^* via:

$$\frac{\{x \cdot a \cdot y\}_{a \in \mathrm{A}}}{x \cdot a_0 \cdot y} \; (A)_{\mathsf{Var}^*}, \tag{2.2}$$

where $x, y \in \mathsf{Var} \setminus \mathrm{supp}(A \cup \{a_0\})$ are distinct.

We define the relation $\vdash_\Gamma \subseteq \wp(\mathsf{Var}^*) \times \mathsf{Var}^*$ to be the smallest relation closed under the following conditions for all $X \subseteq \mathsf{Var}^*$:

- $X \vdash_\Gamma x$ for all $x \in X$,
- If $X \vdash_\Gamma \boldsymbol{\sigma}(uav)$ for all $a \in A$, then $X \vdash_\Gamma \boldsymbol{\sigma}(ua_0v)$, where $[A] = (a_0, A) \in \Gamma$, $u, v \in \mathsf{Var}^*$, and $\boldsymbol{\sigma}$ a substitution.

By Section 1.2.4 we obtain:

Lemma 2.3.1. Let Γ be a set of ISR-equations. Then $\vdash_\Gamma$ is a substitution invariant consequence relation on Var^*.

In this way, we will write $a_0 \leq_\Gamma \bigvee A$ if $A \vdash_\Gamma a_0$, for some finite nonempty $A \subseteq \mathsf{Var}^*$ and term $a_0 \in \mathsf{Var}^*$.

Lemma 2.3.2. Let $\Gamma \cup \{(a_0, A)\}$ be a set of ISR-equations. If $A \vdash_\Gamma a_0$ then $\mathsf{ISR} + \Gamma \models [A]$.

Proof. We induct on the height n of the proof-tree that represents the derivation of a_0 from A. If $n = 0$, then $a_0 \in A$. Since $\vee$ is idempotent, we have $a_0 \vee \bigvee A = \bigvee A$ and hence $a_0 \leq^{\mathsf{ISR}} \bigvee A$. Suppose for every $0 \leq m < n$, if a' has a derivation of height m from A, then $a' \leq^{\mathsf{ISR}+\Gamma} \bigvee A$. Since $n > 0$, there exists a substitution $\boldsymbol{\sigma}$ and $[\mathrm{R}] = (r_0, \mathrm{R}) \in \Gamma$ such that

$$\frac{\{\boldsymbol{\sigma}(xry)\}_{r \in \mathrm{R}}}{\boldsymbol{\sigma}(xr_0y)} \; (\boldsymbol{\sigma}, (\mathrm{R})_{\mathsf{Var}^*}),$$

where $a_0 = \boldsymbol{\sigma}(xr_0y)$ and $\boldsymbol{\sigma}(xry)$ has a derivation of height $m_r < n$ from A for each $r \in \mathrm{R}$. Since $\boldsymbol{\sigma}(xry) \leq_\Gamma \bigvee A$ for each $r \in \mathrm{R}$ and $m_r < n$, it follows by the inductive hypothesis that $\boldsymbol{\sigma}(xry) \leq^{\mathsf{ISR}+\Gamma} \bigvee A$. Hence

$$a_0 \leq_{[\mathrm{R}]} \bigvee_{r \in \mathrm{R}} \boldsymbol{\sigma}(xry) \leq^{\mathsf{ISR}+\Gamma} \bigvee A.$$

Therefore $\mathsf{ISR} + \Gamma \models [A]$. $\square$

2.3.1 The frame $\mathbf{W}_\Sigma$. Define $W := \mathsf{Var}^*$ and $W' := \wp(\mathsf{Var}^*)$. For a fixed set Σ of simple equations, we define $N_\Sigma \subseteq W \times W'$ via

$$x \, N_\Sigma \, X \quad \text{iff} \quad X \vdash_\Sigma x,$$

for all $x \in W$ and $(u, X, v) \in W'$. So (W, W', N_Σ) is a preframe which induces the residuated frame $\mathbf{W}_\Sigma = (W, \widetilde{W'}, \widetilde{N}_\Sigma)$ by Proposition 2.2.2, where we recall that $\widetilde{W'} := W \times W' \times W$ and for all $x \in W$ and $(u, X, v) \in \widetilde{W'}$,

$$x \, \widetilde{N}_\Sigma \, (u, X, v) \quad \text{iff} \quad uxv \, N_\Sigma \, X.$$

Therefore $\mathbf{W}_\Sigma^+$ is a residuated lattice.

Lemma 2.3.3. Let Σ be a set of simple equations. Then $\mathbf{W}_\Sigma^+ \in \mathsf{RL} + \Sigma$.

Proof. If Σ is empty then we are done. So assume Σ is nonempty and let $[\mathrm{R}] = (\mathbf{1_n}, \mathrm{R}) \in \Sigma$. By Proposition 2.2.4, it is enough to show $\mathbf{W} \models (\mathrm{R})_\mathbf{W}$. Let $a_1, \ldots, a_n \in W$ and $(u, X, v) \in \widetilde{W'}$, and suppose $r(\bar{a}) \, \widetilde{N}_\Sigma \, (u, X, v)$ for each $r \in \mathrm{R}$, where $\bar{a} := (a_1, \ldots, a_n)$. By definition, this is equivalent to $u \cdot r(\bar{a}) \cdot v \, N_\Sigma \, X$ for each $r \in \mathrm{R}$, which in turn is equivalent $X \vdash_\Sigma u \cdot r(\bar{a}) \cdot v$ for each $r \in \mathrm{R}$. By definition of $\vdash_\Sigma$, we obtain $X \vdash_\Sigma u \cdot \mathbf{1_n}(\bar{a}) \cdot v$, which is equivalent to $\mathbf{1_n}(\bar{a}) \, \widetilde{N}_\Sigma \, (u, X, v)$. Hence $\mathbf{W}_\Sigma \models (\mathrm{R})_\mathbf{W}$ for each $[\mathrm{R}] \in \Sigma$. Therefore $\mathbf{W}_\Sigma^+ \models \Sigma$ by Proposition 2.2.4. $\square$

Theorem 2.3.4. Let Σ be a set of simple equations. Then for a given proper ISR-equation $[\mathrm{A}] = (a_0, \mathrm{A})$, the following are equivalent:

1. $\mathsf{RL} + \Sigma \models [\mathrm{A}]$.
2. $\mathrm{A} \vdash_\Sigma a_0$.

3. $\mathsf{ISR} + \Sigma \models [\mathrm{A}]$.

Proof. Note that $(2 \Rightarrow 3)$ by Lemma 2.3.2, and $(3 \Rightarrow 1)$ by Proposition 2.1.1. For $(1 \Rightarrow 2)$, suppose $\mathsf{RL} + \Sigma \models [\mathrm{A}]$. Then by Lemma 2.3.3, $\mathbf{W}^+_\Sigma \models [\mathrm{A}]$. By Proposition 2.2.4, this implies $\mathbf{W}_\Sigma \models (\mathrm{A})_{\mathbf{W}}$. Since $\vdash_\Sigma$ is a consequence relation, $\mathrm{A} \vdash_\Sigma a$ for every $a \in \mathrm{A}$, where $[\mathrm{A}] = (a_0, \mathrm{A})$. Hence $a \mathrel{\widetilde{N}_\Sigma} (1, \mathrm{A}, 1)$ for each $a \in \mathrm{A}$. Since $\mathbf{W}_\Sigma \models (\mathrm{A})_{\mathbf{W}}$, this implies $a_0 \mathrel{\widetilde{N}_\Sigma} (1, \mathrm{A}, 1)$, which is equivalent to $\mathrm{A} \vdash_\Sigma a_0$. □

Observe that Theorem 2.3.4 is a partial converse to Proposition 2.1.1, since we only consider simple equations. For instance, if Γ is a set of equations containing a degenerate equation then $\mathsf{RL} + \Gamma \models 1 \leq x$ by Proposition 2.1.2, and so $\mathsf{RL} + \Gamma$ is the trivial variety, while $\mathsf{ISR} + \Gamma$ need not to be. However, there is a stronger relationship between RL and $\mathsf{ISR}_\perp$ in the following way:

Let Γ be a set of ISR-equations and let $\vdash_{\Gamma\perp} \subseteq \wp(\mathsf{Var}^*_\perp) \times \mathsf{Var}^*_\perp$ be the relation closed under

$$\frac{\{x \cdot a \cdot y\}_{a \in \mathrm{A}}}{x \cdot a_0 \cdot y}\ (\mathrm{A})_{\mathsf{Var}^*_\perp},$$

for each $[\mathrm{A}] \in \Gamma$, where $x, y \in \mathsf{Var}_\perp \setminus \mathrm{supp}(\mathrm{A} \cup \{a_0\})$ are distinct. Note that $X \vdash_{\Gamma\perp} \perp$ for every $X \subseteq \mathsf{Var}^*_\perp$.

By the same argument as Lemma 2.3.1, $\vdash_{\Gamma\perp}$ is a substitution invariant consequence relation. By the same argument as Lemma 2.3.2, we obtain:

Lemma 2.3.5. Let $\Gamma \cup \{[\mathrm{A}]\}$ be a set of ISR-equations where $[\mathrm{A}] = (a_0, \mathrm{A})$. Then $\mathrm{A} \vdash_{\Gamma\perp} a_0$ implies $\mathsf{ISR}_\perp \models [\mathrm{A}]$

In a similar fashion to the above, we obtain the preframe $(\mathsf{Var}^*_\perp, \wp(\mathsf{Var}^*_\perp), N_{\Gamma\perp})$ defined via $x \mathrel{N_{\Gamma\perp}} X$ iff $X \vdash_{\Gamma\perp} x$. Let $\mathbf{W}_{\Gamma\perp}$ be the residuated frame induced by this preframe.

Lemma 2.3.6. Let Γ be a set of linear ISR-equations. Then $\mathbf{W}^+_{\Gamma\perp} \in \mathsf{RL} + \Gamma$.

Proof. Observe that $\bot^{\mathbf{W}^+_{\Gamma\bot}} = \{\bot\}$. By same argument as Lemma 2.3.3, we obtain $\mathbf{W}_{\Gamma\bot} \models \Gamma_{\mathbf{W}}$. Therefore, by Lemma 2.2.5, it follows that $\mathbf{W}^+_{\Gamma\bot} \models \Gamma$. □

Theorem 2.3.7. Let $\Gamma \cup \{[\mathrm{A}]\}$ be a set of ISR-equations. Then $\mathsf{ISR}_\bot + \Gamma \models [\mathrm{A}]$ if and only if $\mathsf{RL} + \Gamma \models [\mathrm{A}]$ if and only if $\mathrm{A} \vdash_{\Gamma\bot} a_0$, where $[\mathrm{A}] = (a_0, \mathrm{A})$.

Proof. If Γ contains a degenerate equation, then by Proposition 2.1.2(4), both $\mathsf{ISR}_\bot + \Gamma$ and $\mathsf{RL} + \Gamma$ are the trivial variety, and $\mathrm{A} \vdash_{\Gamma\bot} a_0$ since $\{\bot\} \vdash t$ for all $t \in \mathsf{Var}^*$,[6] so we are done. So let Γ contain no degenerate equations. By Proposition 2.1.2(5) we assume, without loss of generality, that Γ is a set of simple equations. Observe:

$$\begin{aligned} \mathsf{RL} + \Gamma \models [A] \quad &\iff \quad \mathbf{W}^+_{\Gamma\bot} \models [\mathrm{A}] \qquad \text{(Lemma 2.3.6)} \\ &\iff \quad \mathbf{W}_{\Gamma\bot} \models (\mathrm{A})_{\mathbf{W}} \quad \text{(Lemma 2.2.5)} \end{aligned}$$

Now, if $\mathbf{W}_{\Gamma\bot} \models (\mathrm{A})_{\mathbf{W}}$, then $\mathrm{A} \vdash_{\Gamma\bot} a_0$ by definition of $N_{\Gamma\bot}$, and hence $\mathsf{ISR}_\bot + \Gamma \models [\mathrm{A}]$ by Lemma 2.3.5. Conversely, if $\mathbf{W}^+_{\Gamma\bot} \not\models [\mathrm{A}]$, then $\mathsf{ISR}_\bot \not\models [\mathrm{A}]$ since the $\{\vee, \cdot, 1, \bot\}$-reduct of $\mathbf{W}^+_{\Gamma\bot}$ is in $\mathsf{ISR}_\bot + \Gamma$, and hence $\mathrm{A} \nvdash_{\Gamma\bot} a_0$ by Lemma 2.3.5. □

We say a set Γ of ISR-equations is *degenerate* if it contains a degenerate equation. For Γ not degenerate, the *simplification* of Γ is the set Σ_Γ containing all the equivalent simple equations given by Proposition 2.1.2.

Corollary 2.3.8. Let Γ be a set of ISR-equations. Then $\mathsf{RL} + \Gamma$ is the trivial variety if and only if Γ is degenerate.

---

[6] If $[\mathrm{R}] = (r_0, \mathrm{R})$ is degenerate, then there exists $x_{i_r} \in \mathrm{supp}(r) \setminus \mathrm{supp}(r_0)$ for each $r \in \mathrm{R}$. Let $\overline{u} \in \overline{\mathsf{Var}^*_\bot}$ such that

$$u_i = \begin{cases} \bot & \text{if } i = i_r \text{ for some } r \in \mathrm{R} \\ 1 & \text{otherwise} \end{cases} .$$

Then $t \cdot r(\overline{u}) = t \cdot \bot = \bot$ for each $r \in \mathrm{R}$ and $t \cdot r_0(\overline{u}) = t \cdot 1 = t$. Hence $\{\bot\} \vdash_{[\mathrm{R}]\bot} t$. Then $\mathrm{A} \vdash_{[\mathrm{R}]\bot} t$ since $\mathrm{A} \vdash_{[\mathrm{R}]\bot} \bot$.

Proof. The reverse direction follows from Proposition 2.1.2(4). For the forward direction, note $\mathbf{W}^+_{\Gamma\bot} \in \mathsf{RL}+\Gamma$ by Lemma 2.3.6, so if $\mathsf{RL}+\Gamma$ is the trivial variety then $\mathbf{W}^+_{\Gamma\bot} \models 1 \leq x$. Hence $\mathbf{W}_{\Gamma\bot} \models (1 \leq x)_{\mathbf{W}}$ by Proposition 2.2.3, and thus $\{x\} \vdash_{\Gamma\bot} 1$. We proceed by inspecting the proof-tree of $\{x\} \vdash_{\Gamma\bot} 1$, which we may assume is of minimal height. By definition, there exists a substitution $\boldsymbol{\sigma}$, $u, v \in \mathsf{Var}^*_\bot$, and $[\mathrm{A}] = (a_0, \mathrm{A}) \in \Gamma$ such that $1 = u\boldsymbol{\sigma}(a_0)v$, $\{x\} \vdash_{\Gamma\bot} u\boldsymbol{\sigma}(a)v$ for all $a \in \mathrm{A}$, and

$$\frac{\{u\boldsymbol{\sigma}(a)v\}_{a\in\mathrm{A}}}{u\boldsymbol{\sigma}(a_0)v}\ (\boldsymbol{\sigma}, (\mathrm{R})_{\mathsf{Var}^*_\bot}).$$

Now, since $1 = u\boldsymbol{\sigma}(a_0)v$, it follows that $u = v = 1$ and for every $x_i \in \mathrm{supp}(a_o)$, $\boldsymbol{\sigma}(x_i) = 1$. Let $a \in \mathrm{A}$. Since the proof is of minimal height, it must be that $u\boldsymbol{\sigma}(a)v \neq 1$. So there is $x_j \in \mathrm{supp}(a)$ such that $\boldsymbol{\sigma}(x_j) \neq 1$, and thus $\mathrm{supp}(a) \setminus \mathrm{supp}(a_0) \neq \emptyset$. Therefore $[\mathrm{A}]$ is degenerate. □

Theorem 2.3.9. If Γ is a non-degenerate set of ISR-equations, then $\mathsf{RL} + \Gamma \models [\mathrm{A}]$ if and only if $A \vdash_\Sigma a_0$, where $[\mathrm{A}] = (a_0, \mathrm{A})$ is an ISR-equation and $\Sigma = \Sigma_\Gamma$ is the simplification of Γ.

Proof. By Proposition 2.1.2(5), $\mathsf{RL} + \Sigma = \mathsf{RL} + \Gamma$ and, by Theorem 2.3.7, $\mathsf{RL} + \Sigma \models [\mathrm{A}]$ iff $\mathrm{A} \vdash_{\Sigma\bot} a_0$. Observe that the reverse direction follows by Theorem 2.3.7 since $\mathrm{A} \vdash_\Sigma a_0$ implies $\mathrm{A} \vdash_{\Sigma\bot} a_0$. For the forward direction, assume $\mathsf{RL} + \Sigma \models [\mathrm{A}]$. We induct on the height k of the proof-tree for $\mathrm{A} \vdash_{\Sigma\bot} a_0$, which we may assume is of minimal height. If the height is $k = 0$ then $[\mathrm{A}] \in \Sigma$. This implies $\mathrm{A} \vdash_\Sigma a_0$ by definition. Suppose the claim holds for all $\vdash_{\Sigma\bot}$-proofs of height less than $k > 0$. Then there exists a substitution $\boldsymbol{\sigma}$ and $[\mathrm{R}] = (\mathbf{1_n}, \mathrm{R}) \in \Sigma$ such that $\mathrm{A} \vdash_{\Sigma\bot} \{x\boldsymbol{\sigma}(r)y : r \in \mathrm{R}\}$ and $x\boldsymbol{\sigma}(\mathbf{1_n})y = a_0$, for some $x, y \in \mathsf{Var}^*_\bot$. By the inductive hypothesis, $\mathrm{A} \vdash_\Sigma \{x\boldsymbol{\sigma}(r)y : r \in \mathrm{R}\}$. So $x\boldsymbol{\sigma}(r)y \neq \bot$, for all $r \in \mathrm{R}$, and hence $x, y \neq \bot$ and $\boldsymbol{\sigma}(x_i) \neq \bot$ for all $1 \leq i \leq n$. Hence $\{x\boldsymbol{\sigma}(r)y : r \in \mathrm{R}\} \vdash_\Sigma x\boldsymbol{\sigma}(\mathbf{1_n})y = a_0$. Therefore $\mathrm{A} \vdash_\Sigma a_0$. □

2.4 Knotted and other special ISR-equations

Given $t \in \mathsf{Var}^*$ and $x \in \mathsf{Var}$, by $\#(t,x)$ we denote the *length of* x *in* t, where $\#(t,x)$ the number of occurrences of the variable x in t. By $\#(t)$ we denote the *length of* t to be to total number symbols in t, i.e., $\#(t) = \sum_{x \in \mathrm{supp}(t)} \#(t,x)$.

Definition 2.4.1. Let $[A]$ be an ISR-equation. We say $[A]$ is:

- *knotted* if $[A] : x^n \leq x^m$ for some $n \neq m$.
- *expansive* if $[A] : x^n \leq \bigvee_{p \in P} x^{n+p}$, for some $n > 0$ and finite nonempty $P \subseteq \mathbb{Z}^+$.
- *compressive* if $[A] : x^n \leq \bigvee_{p \in P} x^p$, for some $n > 0$ and nonempty $P \subseteq \{1, ..., n-1\}$.
- k-*mingle* if $[A] : x^k \leq x$, for some $k > 1$.

A simple equation $[\mathrm{R}]$ is called *pre-knotted*, *pre-expansive*, or *pre-compressive* if there exists a substitution $\boldsymbol{\sigma}$ such that $[\boldsymbol{\sigma}\mathrm{R}]$ is knotted, expansive, or compressive, respectively. We say $[\mathrm{R}]$ is *mingly* if $[\mathrm{R}]$ is integral or there is a substitution $\boldsymbol{\sigma}$ such that $[\boldsymbol{\sigma}\mathrm{R}]$ is k-mingle for some $k > 1$. We say a set Σ of simple equations has a property if it contains an equation with that same property.

We say a variety $\mathcal{V} \subseteq \mathsf{RL}$ is *knotted*, *expansive*, *compressive*, or *mingly* if $\mathcal{V} \models [A]$ for some equation $[A]$ that is knotted, expansive, compressive, or mingly, respectively. We denote the knotted equation $x^n \leq x^m$ by $[\mathsf{k}_n^m]$, and by $[\mathsf{K}_n^m]$ we denote the linearization of $[\mathsf{k}_n^m]$, i.e. $[\mathsf{K}_n^m] = (\mathbf{1_n}, K_n^m)$, where $K_n^m := \{t \in \{x_1, \ldots, x_n\}^* : \#(t) = m\}$. Note that by Proposition 2.1.2, $[\mathsf{k}_n^m]$ and $[\mathsf{K}_n^m]$ are RL-equivalent.

Theorem 2.4.1. Let Σ be a set of simple equations.

1. $\mathsf{RL} + \Sigma$ is integral iff Σ is integral.
2. $\mathsf{RL} + \Sigma$ is knotted iff Σ is pre-knotted.
3. $\mathsf{RL} + \Sigma$ is expansive iff Σ is pre-expansive.

4. $\mathsf{RL} + \Sigma$ is compressive iff Σ is pre-compressive.

5. $\mathsf{RL} + \Sigma$ is mingly iff Σ is mingly.

Proof. Note that the reverse direction clearly follows for each case. The forward direction is of the form $\mathsf{RL} + \Sigma \models [A]$ for some ISR-equation $[A] = (a_0, A)$ for each case, and hence $A \vdash_\Sigma a_0$ by Theorem 2.3.9. We proceed by inspecting the leaves of its proof-tree of minimal height for each case. Note that a leaf must be of the form:

$$\frac{\{u\boldsymbol{\sigma}(r)v\}_{r\in \mathrm{R}}}{u\boldsymbol{\sigma}(\mathbf{1_k})v}\ (\boldsymbol{\sigma}, (\mathrm{R})),$$

where $\{u\boldsymbol{\sigma}(r)v : r \in \mathrm{R}\} \subseteq A$, for some substitution $\boldsymbol{\sigma}$, $[\mathrm{R}] = (\mathbf{1_k}, \mathrm{R}) \in \Sigma$, and $u, v \in \mathsf{Var}^*$. Since the proof is of minimal height, $u\boldsymbol{\sigma}(\mathbf{1_k})v \neq u\boldsymbol{\sigma}(r)v$ and hence $\boldsymbol{\sigma}(\mathbf{1_k}) \neq \boldsymbol{\sigma}(r)$, for each $r \in \mathrm{R}$.

(1) Suppose $[A] : x \leq 1$ is integrality. Then $a_0 = x$ and $A = \{1\}$. So $u\boldsymbol{\sigma}(r)v = 1$ for each $r \in \mathrm{R}$, and hence $u = v = \boldsymbol{\sigma}r = 1$. Thus $\boldsymbol{\sigma}(x_i) = 1$ for each $x_i \in \mathrm{supp}(\mathrm{R})$. Now, since $\boldsymbol{\sigma}(\mathbf{1_k}) \neq 1$, there must exist $x_j \in \mathrm{supp}(\mathbf{1_k})$ such that $\boldsymbol{\sigma}(x_j) \neq 1$. But this implies $x_j \in \mathrm{supp}(\mathbf{1_k}) \setminus \mathrm{supp}(\mathrm{R})$, and so $[\mathrm{R}]$ is integral.

(2) Suppose $[A] : x^n \leq x^m$ is knotted. If $[\mathrm{R}]$ is integral then it is pre-knotted and we are done, so we assume $\mathrm{supp}(\mathrm{R}) = \mathrm{supp}(\mathbf{1_k})$. Now, $a_0 = x^n$ and $A = \{x^m\}$ for some $n \neq m$. Hence $u\boldsymbol{\sigma}(r)v = x^m$ for each $r \in \mathrm{R}$, so $\boldsymbol{\sigma}[\mathrm{R}] = \{x^c\}$ for some $c \leq m$. Since $\boldsymbol{\sigma}(x_i) = x$ for each $x_i \in \mathrm{supp}(\mathrm{R})$, It follows that $\boldsymbol{\sigma}(\mathbf{1_k}) = x^d$ for some $d \geq 0$. Hence $c \neq d$ since $\boldsymbol{\sigma}(\mathbf{1_k}) \neq \boldsymbol{\sigma}(r)$. But this implies $[\boldsymbol{\sigma}\mathrm{R}] = x^c \leq x^d$ for some $c \neq d$, a knotted rule. Hence $[\mathrm{R}]$ is pre-knotted.

(3) [(4)] Suppose $[A] : x^n \leq \bigvee_{p\in P} x^{n+p}$ is expansive [compressive], for some $n > 1$ and finite nonempty $P \subseteq \mathbb{Z}^+$ [$P \subseteq \{1, \ldots, n-1\}$]. By the same argument in (2), for each leaf $(\boldsymbol{\sigma}, (\mathrm{R}))$, $\boldsymbol{\sigma}(\mathbf{1_k}) = x^{c_\mathrm{R}}$ for some $c_\mathrm{R} \geq 0$. If there exists a leaf $(\boldsymbol{\sigma}, (\mathrm{R}))$ such that $c_\mathrm{R} < d$ [$c_\mathrm{R} > d$] for all $x^d \in \boldsymbol{\sigma}[\mathrm{R}]$, then $[\boldsymbol{\sigma}\mathrm{R}]$ is expansive [compressive], and we are

done. Otherwise, for every leaf $(\boldsymbol{\sigma}, (\mathrm{R}))$, since $\boldsymbol{\sigma}(\mathbf{1_k}) \notin \boldsymbol{\sigma}[\mathrm{R}]$ it follows that $c_{\mathrm{R}} > d_{\mathrm{R}} > n$ [$c_{\mathrm{R}} < d < n$] for some $x^{d_{\mathrm{R}}} \in \boldsymbol{\sigma}[\mathrm{R}]$. So $[A'] : x^n \leq \bigvee_{r \in \mathrm{R}} x^{c_{\mathrm{R}}}$ is an expansive [compressive] equation such that $A' \vdash_\Sigma x^n$, where each branch of its proof-tree has height strictly less than each branch in the proof-tree of $A \vdash_\Sigma x^n$. Continuing this process inductively, we conclude Σ contains a pre-expansive [pre-compressive] equation.

(5) We may assume Σ is not integral, otherwise we are done by (1). So suppose $[A] : x^n \leq x$ is n-mingle for some $n > 1$. Then $a_0 = x^n$ and $A = \{x\}$. So $u\boldsymbol{\sigma}(r)v = x$ for each $r \in \mathrm{R}$. Note that $u = x$ or $v = x$ implies $\boldsymbol{\sigma}(r) = 1$, and since $[\mathrm{R}]$ is not integral, $\mathrm{supp}(\mathrm{R}) = \mathrm{supp}(\mathbf{1_k})$ and so it follows that $\boldsymbol{\sigma}(\mathbf{1_k}) = 1$. Since $\boldsymbol{\sigma}(\mathbf{1_k}) \neq \boldsymbol{\sigma}(r)$ it must be that and hence $u = v = 1$. Thus $\boldsymbol{\sigma}(r) = x$ for all $r \in \mathrm{R}$. Hence for every $r \in \mathrm{R}$, there exists $x_r \in \mathrm{supp}(r)$ such that $\boldsymbol{\sigma}(x_r) = x$ and $\boldsymbol{\sigma}(y) = 1$ for all $y \in \mathrm{supp}(r) \setminus \{x_r\}$. Hence $\boldsymbol{\sigma}(\mathbf{1_k}) = x^m$ for some $m \geq 1$. Since $\boldsymbol{\sigma}\mathbf{1_k} \neq x$, it follows that $m > 1$. Hence $[\boldsymbol{\sigma}\mathrm{R}]$ is m-mingle, and therefore $[\mathrm{R}]$ is mingly. Thus $\boldsymbol{\sigma}(x_i) = 1$ for each $x_i \in \mathrm{supp}(\mathrm{R})$. Now, since $\boldsymbol{\sigma}(\mathbf{1_k}) \neq 1$, there must exist $x_j \in \mathrm{supp}(\mathbf{1_k})$ such that $\boldsymbol{\sigma}(x_j) \neq 1$. But this implies $x_j \in \mathrm{supp}(\mathbf{1_k}) \setminus \mathrm{supp}(\mathrm{R})$, and so $[\mathrm{R}]$ is integral. □

Let $0 < n < m \in \mathbb{N}$. We say a variety $\mathcal{V} \subseteq \mathsf{RL}$ is (n, m)-*potent* if $\mathcal{V} \models x^n = x^m$, and we will say the $\mathcal{V}$ is *potent* if it is (n, m)-potent for some $0 < n < m$. We say a pre-knotted equation $[A]$ is *expansive* [resp. *compressive*] if the knotted equation witnessing that $[A]$ is pre-knotted is expansive [compressive].

Lemma 2.4.2. Let $[\mathsf{k}^a_{a+c}]$ and $[\mathsf{k}^{b+d}_b]$ be compressive and expansive knotted rules, respectively, for some $a, b, c, d > 0$. Then $\mathsf{RL} + [\mathsf{k}^a_{a+c}] + [\mathsf{k}^{b+d}_b]$ is $(n, n + m)$-potent, where $n = \max(a, b)$ and $m = \min(c, d)$.

Proof. On the one hand, if $n = a$, then $x^{n+c} \leq_{[\mathsf{k}^a_{a+c}]} x^n$ by definition. Since $b \leq n$ and multiplication is order-preserving, it follows that $x^n \leq_{[\mathsf{k}^{b+d}_b]} x^{n+d}$. On the other hand, if $n = b$, then $x^n \leq_{[\mathsf{k}^{b+d}_b]} x^{n+d}$ by definition, and since $a \leq n$, it follows that $x^{n+c} \leq_{[\mathsf{k}^a_{a+c}]} x^n$.

In either case, we find that $x^{n+c} \leq_\Sigma x^n \leq_\Sigma x^{n+d}$, where $\Sigma = \{[\mathsf{k}^a_{a+c}], [\mathsf{k}^{b+d}_b]\}$. Applying $[\mathsf{k}^a_{a+c}]$ on the left and $[\mathsf{k}^{b+d}_b]$ on the right, we obtain:[7]

$$x^{n+\mathrm{lcm}(c,d)} \leq_{[\mathsf{k}^a_{a+c}]} x^{n+c} \leq_{[\mathsf{k}^a_{a+c}]} x^n \leq_{[\mathsf{k}^{b+d}_b]} x^{n+d} \leq_{[\mathsf{k}^{b+d}_b]} x^{n+\mathrm{lcm}(c,d)}.$$

Hence $\mathsf{RL} + \Sigma \models x^n = x^{n+m}$. □

Corollary 2.4.3. Let Σ be a set of simple equations and $\mathcal{V} \in \{\mathsf{RL}, \mathsf{FL}\}$. Then $\mathcal{V}+\Sigma$ is potent if and only if Σ is pre-compressive [or resp. pre-expansive] and contains an expansive [resp. compressive] pre-knotted equation.

Proof. By Proposition 2.1.2, $\mathcal{V} + \Sigma$ is (n, m)-potent if and only if $\mathsf{RL} + \Sigma \models \{[\mathsf{k}^m_n], [\mathsf{k}^n_m]\}$. Since $[\mathsf{k}^m_n]$ and $[\mathsf{k}^n_m]$ are expansive and compressive knotted equations, respectively, it follows that $\mathsf{RL} + \Sigma \models \{[\mathsf{k}^m_n], [\mathsf{k}^n_m]\}$ implies Γ is pre-knotted, pre-expansive, and pre-compressive by Theorem 2.4.1, satisfying the forward implication. For the reverse, suppose Σ is pre-knotted, pre-expansive, and pre-compressive by Theorem 2.4.1. Since the other case can be handled similarly, without loss of generality we may assume the witnesses are an expansive $[E]$ and compressive knotted $[\mathsf{k}^a_{a+c}]$, where $[E] : x^n \leq \bigvee_{p\in P} x^{n+p}$ for some $a, c, n > 0$ with finite nonempty $P \subseteq \mathbb{Z}^+$. By Lemma 2.4.2, it is enough to show that $\mathsf{RL} + \Sigma \models [\mathsf{k}^{b+d}_b]$ for some $b, d > 0$. Let $N = \max\{a, n\}$, $b = cN$ and $d = c$. Then $x^{c(N+k)} \leq_{[\mathsf{k}^a_{a+c}]} x^{c(N+1)} \leq_{[\mathsf{k}^a_{a+c}]} x^{cN}$ for all $k \geq 1$. Hence, since $p > 0$ for every $p \in P$,

$$x^b = (x^c)^N \leq_{[E]} \bigvee_{p\in P} x^{c(N+p)} \leq_{[\mathsf{k}^a_{a+c}]} x^{cN+c} \leq_{[\mathsf{k}^a_{a+c}]} x^b.$$

So $\mathsf{RL} + \Sigma \models [\mathsf{k}^{b+d}_b]$. Therefore $\mathsf{RL} + \Sigma$ is $(b, b + d)$-potent. □

[7] Where $\mathrm{lcm}(j, k)$ denotes the *least common multiple* of integers $j, k \in \mathbb{N}$.

2.5 Deduction theorem for expansive varieties

In certain cases, the satisfaction of a quasi-equation can be related the satisfaction a single equation. If the satisfaction of a quasi-equation is equivalent to the satisfaction of an equation, for all quasi-equations, we say that variety has a *deduction theorem*. The existence of a deduction theorem can be vitally useful, in particular, for establishing decidability results. For instance, in Chapter 3 we will use a deduction theorem to establish that the quasi-equational theory for some varieties are decidable using the fact that their equational theory has a decision procedure, while in Chapters 4 and 5 we will use a deduction theorem to establish the undecidability of the equational theory for some varieties by using the undecidability of their quasi-equational theory. In this section we will demonstrate that all expansive varieties have a deduction theorem. First we must review some preliminary notions.

The negative cone of a residuated lattice $\mathbf{A}$ is the set $A^- = \{a \in A : a \leq 1\}$. We say that a variety $\mathcal{V} \subseteq \mathsf{CRL}$ is *negatively n-potent* if the negative cone of each algebra in $\mathcal{V}$ is n-potent, i.e., $\mathcal{V} \models (x \wedge 1)^n = (x \wedge 1)^{n+1}$ (or equivalently, $\mathcal{V} \models (x \wedge 1)^n \leq (x \wedge 1)^{n+1}$).

Let t be a term and S be a finite set of terms in the language of CRL. It can be easily verified that

$$\begin{array}{rcl} (\exists m \in \mathbb{N})(\exists s_1, \ldots, s_m \in S)\ \mathsf{CRL} & \models & \prod_{i=1}^{m}(1 \wedge s_i) \leq t \\ \text{if and only if} \quad (\exists k \in \mathbb{N})\ \mathsf{CRL} & \models & (1 \wedge \bigwedge S)^k \leq t. \end{array} \tag{2.3}$$

Clearly the forward direction is satisfied by taking $k = m$, since $s \geq \bigwedge S$, for all $s \in S$. The reverse direction holds by setting $m = k \cdot |S|$, and observing that $\prod_{s \in S}(s \wedge 1) \leq 1 \wedge \bigwedge S$.

If $\mathcal{V} \subseteq \mathsf{CRL}$ is a negatively n-potent variety, then we obtain

$$(\exists m \in \mathbb{N})(\exists s_1, \ldots, s_m \in S)\ \mathcal{V} \models \prod_{i=1}^{m}(1 \wedge s_i) \leq t \iff \mathcal{V} \models (1 \wedge \bigwedge S)^n \leq t, \tag{2.4}$$

where reverse direction follows from Equation (2.3), while the forward direction uses the fact that $(1 \wedge x)^n \leq (1 \wedge x)^k$, if $k \leq n$, and $(1 \wedge x)^n = (1 \wedge x)^k$, if $k > n$, by the negative n-potency of $\mathcal{V}$.

We consider the quasi-equation $\xi_S(t)$ and the equation $\epsilon^n_S(t)$, respectively, below:

$$\underset{s \in S}{\&} 1 \leq s \implies 1 \leq t \qquad\qquad (1 \wedge \bigwedge S)^n \leq t.$$

In this way we establish the fact that satisfaction of quasi-equations in a negatively n-potent subvariety of CRL is equivalent to the satisfaction of a corresponding equation.

Theorem 2.5.1. If $\mathcal{V}$ is a negatively n-potent subvariety of CRL and $S \cup \{t\}$ a finite set of terms in the language of $\mathcal{V}$, then

$$\mathcal{V} \models \xi_S(t) \iff \mathcal{V} \models \epsilon^n_S(t).$$

Proof. Let $\mathbf{F}_\mathcal{V}$ be the free algebra for $\mathcal{V}$ over countably many generators, and define the congruence $C := \mathrm{Cg}(\{(1 \wedge s, s) : s \in S\})$. We denote the quotient algebra of C on $\mathbf{F}_\mathcal{V}$ by $\mathbf{F}_\mathcal{V}/C$. For a subset X of $\mathrm{F}^-_\mathcal{V}$, we denote by $M(X)$ the convex normal submonoid of $\mathrm{F}^-_\mathcal{V}$ generated by X.[8] Observe that $\mathcal{V} \models \underset{s \in S}{\&} 1 \leq s \Rightarrow 1 \leq t$

$$\begin{aligned}
&\iff && \text{in } \mathbf{F}_\mathcal{V}/C,\ [1 \wedge t]_C = [1]_C && \\
&\iff && \text{in } \mathbf{F}_\mathcal{V},\ (1 \wedge t) \in M(\{1 \wedge s : s \in S\}) && [9] \\
&\iff && \text{in } \mathbf{F}_\mathcal{V},\ (\exists m \in \mathbb{N})(\exists s_1, \ldots, s_m \in S)\ \textstyle\prod_{i=1}^m (1 \wedge s_i) \leq t && [9] \\
&\iff && (\exists m \in \mathbb{N})(\exists s_1, \ldots, s_m \in S)\ \mathcal{V} \models \textstyle\prod_{i=1}^m (1 \wedge s_i) \leq t && \\
&\iff && \mathcal{V} \models (1 \wedge \bigwedge S)^n \leq t && \text{Eq. (2.4).}
\end{aligned}$$

[8] See Theorem 3.47 in [9].

□

If $[E] : x^n \leq \bigvee_{p \in P} x^{n+p}$ is an expansive equation, then $\mathsf{CRL} + [E]$ is negatively n-potent since $x \wedge 1 \leq 1$ and thus $(x \wedge 1)^{n+k} \leq (x \wedge 1)^{n+1} \leq (x \wedge 1)^n$ for every $k \geq 1$, which in the presence of $[E]$ implies $(x \wedge 1)^n \leq \bigvee_{p \in P} (x \wedge 1)^{n+p} \leq (x \wedge 1)^{n+1} \leq (x \wedge 1)^n$, i.e., $(x \wedge 1)^n = (x \wedge 1)^{n+1}$.

Corollary 2.5.2. $\mathsf{CRL} + \Gamma$ admits a deduction theorem for every pre-expansive set Γ of ISR-equations, and thus the computational complexity for its equational theory is at least as complex as the complexity of its quasi-equational theory.

Chapter 3: Decidability and Complexity Upper-bounds

In this chapter we establish the decidability of many structures extended by the equations and structural rules presented in Chapter 2. In the first section, we show how [3] establishes the failure of the finite embeddability property for a collection of special simple equations that are satisfied by (products of) chains. In the second section, we provide some sufficient conditions that guarantee the $\{\vee, \cdot, 1\}$-fragment of the equational theory for certain subvarieties of RL are decidable using Theorem 2.3.4. The third section utilizes results in [8] for proving the finite model property for subvarieties of RL extended by so-called *completely linear equations*. In the last section, we give proof-theoretic decision procedure for potent-varieties. This is a generalization of the proof due to Gentzen [11] showing the decidability of the Gentzen-system $\mathbf{FL}_{\mathsf{ecw}}$ for propositional intuitionistic logic. Furthermore, we show that this decision procedure is at worst double-exponential with respect to the number of symbols present in the input.

3.1 The FMP, FEP, and some known results

A class of algebras $\mathcal{K}$ is said to have the *finite model property* (FMP) if every equation that fails in $\mathcal{K}$ fails in a finite member of $\mathcal{K}$. As a consequence of Harrop's theorem (see [9]), if $\mathcal{K}$ is finitely axiomatizable, of finite type, and has the FMP, then $\mathcal{K}$ has a decidable equational theory.

We say a class $\mathcal{K}$ of algebras has the *finite embeddability property* (FEP) when for any given finite partial subalgebra $\mathbf{B}$ of an algebra $\mathbf{A}$ in $\mathcal{K}$, there exists a finite algebra $\mathbf{D}$ in $\mathcal{K}$ into which $\mathbf{B}$ can be embedded. In particular, $\mathcal{K}$ is generated by its finite members, and therefore has the FMP. Furthermore, if $\mathcal{K}$ is a quasivariety of finite type (e.g., $\mathcal{K} \subseteq \mathsf{RL}$ or $\mathcal{K} \subseteq \mathsf{FL}$), the FEP is equivalent to the *strong finite model property*, i.e., every quasi-identity

that fails in $\mathcal{K}$ fails on a finite member of $\mathcal{K}$. Consequently, if $\mathcal{K}$ is finitely axiomatizable, of finite type, and has the FEP, then its universal theory, and quasi-equational theory in particular, are decidable (see [9]).

In [24], van Alten establishes that $\mathsf{CRL} + [\mathsf{k}_n^m]$ has the FEP, for any knotted equation $[\mathsf{k}_n^m]$. Furthermore, the residuated frames construction in [8] demonstrates:

Proposition 3.1.1 ([24],[8])**.** $\mathsf{CRL} + \Gamma$ has the FEP for any pre-knotted set Γ of ISR-equations.

3.1.1 Failure of the FEP. In [3], Blok and van Alten track the failure of the FEP down to the existence of a certain infinite algebra.[1] Clearly, structure $\mathbb{Z}$ with its natural ordering as a chain, product as integer addition, and residuation as integer subtraction, is a commutative residuated lattice and it falsifies the quasi-equation

$$x \geq 1 \;\&\; xy = 1 \implies x = 1, \tag{3.1}$$

which says that the only positive invertible element is the unit.[2] However, this quasi-equation is satisfied in every finite commutative residuated lattice.[3] The same argument works with $\mathbb{Z}$ expanded with an additional constant 0, set to be any element of $\mathbb{Z}$.

Proposition 3.1.2 ([3])**.** Any subvariety of RL or FL containing $\mathbb{Z}$ lacks the FEP.

We call a residuated lattice *representable* (or *semilinear*) if it is the subdirect product of chains. As shown by Hart, Rafter, and Tsinakis in [12], a commutative residuated lattice

[1] Specifically, a lattice-ordered abelian group, or abelian l-group.

[2] Written the standard notation of $\mathbb{Z}$, the quasi-equation is read $x \geq 0 \;\&\; x + y = 0 \implies x = 0$, which is clearly false in $\mathbb{Z}$.

[3] If $\mathbf{A} \in \mathsf{CRL}$ is finite, then it is bounded. If $x > 1$ is invertible then $x^2 > x$ and x^2 is invertible, since $1 \leq x$ implies $x \leq x^2$ and $xy = 1$ implies $1 = xy \leq x^2 y = x$. Hence $\top = x^n$ for some $n \in \mathbb{N}$ is invertible. But this results in a contradiction since $\top y = 1$ implies $\top \leq 1$ since $\top^2 = \top$.

is representable iff it satisfies the equation

$$1 \leq (x \to y) \vee (y \to x) \quad \text{(prelinearity)}.$$

Since $\mathbb{Z}$ is a chain, if $\mathcal{V} \subseteq \mathsf{CRL}$ contains the variety of prelinear commutative residuated lattices, then $\mathbb{Z} \in \mathcal{V}$, and hence $\mathcal{V}$ lacks the FEP.

Proposition 3.1.3. If a set Σ of ISR-equations is a CRL-consequence of prelinearity, then $\mathsf{CRL} + \Sigma$ lacks the FEP. In particular, $\mathsf{CRL} + [A]$ lacks the FEP for

$$[A] : s^m t^n \leq s^{2m} \vee t^{2n},$$

where $s, t \in \mathsf{Var}^*$ and $m, n \in \mathbb{N}$.

Proof. Prelinearity implies $1 \leq (t^n \to s^m) \vee (s^m \to t^n)$. Since multiplication is order-preserving and distributes over joins, this implies

$$s^m t^n \leq s^m t^n (t^n \to 1) \vee s^m t^n (1 \to t^n) \leq s^{2m} \vee t^{2n}.$$

Hence $[A]$ is satisfied by any chain in CRL. Therefore $\mathsf{CRL} + [A]$ lacks the FEP. □

3.2 A note on decidability in ISR

For a set of simple equations Σ, Theorem 2.3.9 shows that the equational theories for the $\mathsf{ISR} + \Sigma$ and the $\{\vee, \cdot, 1\}$-reduct of $\mathsf{RL} + \Sigma$ are equivalent. Therefore any decision procedure for one is a decision procedure for the other. E.g., by Proposition 3.1.1,

Theorem 3.2.1. If Σ is a pre-knotted set of simple equations, then $\mathsf{ISR} + \Sigma$ has the FMP.

On the other hand, suppose Σ is a set of simple equations for which given any finite set $A \subseteq \mathsf{Var}^*$, the Σ-closure $\Gamma_\Sigma(A) := \{x \in \mathsf{Var}^* : A \vdash_\Sigma x\}$ is finite. Call such a set

Σ *downwards-finite*. For example, consider the equation $[\mathrm{R}] : x \leq x^2 \vee 1$. For any finite set $A \subseteq \mathsf{Var}^*$, $t \in \Gamma_{[\mathrm{R}]}(A)$ implies $\mathrm{supp}(t) \subseteq \mathrm{supp}(A)$ and the degree of t is no larger than the degree of A, i.e., $\#(t) \leq \max\{\#(a) : a \in A\}$. Since $\mathrm{supp}(A)$ is finite, the set $T_A := \{t \in \mathsf{Var}^* : \mathrm{supp}(t) \subseteq \mathrm{supp}(A) \;\&\; \#(t) \leq \#(A)\}$ is finite, establishing that $\Gamma_{[\mathrm{R}]}(A)$ is finite since $\Gamma_{[\mathrm{R}]}(A) \subseteq T_A$. Hence $[\mathrm{R}]$ is downwards-finite.

Now by Theorem 2.3.9, $\mathsf{ISR}{+}[\mathrm{R}] \models [A]$ if and only if $A \vdash_{[\mathrm{R}]} a_0$, for any non-degenerate $[A]$ where (a_0, A) is the simplification of $[A]$. Let $\gamma_{[\mathrm{R}]}(X)$ be the single step closure of a set $X \subseteq \mathsf{Var}^*$ by $(\mathrm{R})^*_{\mathsf{Var}}$ (see Equation (2.2)). It is easily verified that $\Gamma_{[\mathrm{R}]}(X) = \bigcup_{n\in\mathbb{N}} \gamma^n_{[\mathrm{R}]}(X)$. Since A is finite and $[\mathrm{R}]$ is downwards-finite, there is an $n_A \in \mathbb{N}$ such that $\gamma^{n_A}_{[\mathrm{R}]}(A) = \Gamma_{[\mathrm{R}]}(A)$. Thus for all $t \in \mathsf{Var}^*$, $\mathsf{ISR} + [\mathrm{R}] \models t \leq \bigvee A$ iff $t \in \gamma^{n_A}_{[\mathrm{R}]}(A)$. This a decision procedure for the equational theory of $\mathsf{ISR} + [\mathrm{R}]$. By the same argument,[4]

Theorem 3.2.2. Let Σ be a finite set of simple equations. If Σ is downwards-finite then the equational theories of $\mathsf{ISR} + \Sigma$ and the $\{\vee, \cdot, 1\}$-reduct of $\mathsf{RL} + \Sigma$ are decidable.

The set Σ being downwards-finite is only a sufficient condition for decidability. Indeed, extensions of ISR by compressive knotted rules $[\mathsf{k}^n_{n+m}]$ (and thus also their linearization $[\mathsf{K}^n_{n+m}]$) have a decidable equational theory, even though they are not downwards-finite. For instance, consider the set $A = \{x^n\}$. Then $\Gamma_{[\mathsf{K}^n_{n+m}]}(A) \supseteq \{x^{n+km} : k \in \mathbb{N}\}$, an infinite set. We note that there are no examples known to the author for which the equational theory of $\mathsf{ISR} + [\mathrm{R}]$ is undecidable.

3.3 The FMP and completely linear simple equations

Let $\mathcal{L}$ be a substructural logic. For a sequent s in $\mathcal{L}$, we define $s^{\leftarrow}$ to be the set of all sequents involved in an exhaustive proof search for s. Precisely, $s^{\leftarrow}$ is the least set of sequents such that $s \in s^{\leftarrow}$ and if (t, T) is an instance of a rule of $\mathcal{L}$ and $t \in s^{\leftarrow}$, then $T \subseteq s^{\leftarrow}$. Clearly $s^{\leftarrow}$ is the set of all sequents involved in an exhaustive proof search for

[4] In the case that $|\Sigma| = n \geq 2$, index $\Sigma = \{[\mathrm{R}_1], \ldots, [\mathrm{R}_n]\}$ and define $\gamma_\Sigma := \gamma_{[\mathrm{R}_n]} \circ \cdots \circ \gamma_{[\mathrm{R}_1]}$.

s. We say that a rule (r) in $\mathcal{L}$ *does not increase complexity* if for each instance of the rule, the complexity of each sequent in the numerator is no larger than the complexity of the denominator. By *complexity*, we typically mean a function from the set of sequents to some well partially-ordered set.[5] For some structural rules, complexity for a sequent can be defined to be, for example, its *length*, i.e., the number of symbols which occur. However, rules like contraction (c) or the cut-rule $(\mathbf{cut})$, are examples of structural rules which do increase complexity. We note, though, that extensions $\mathbf{FL}$ by simple rules enjoy cut-admissibility, and so one only need consider the set $s^{\leftarrow}$ omitting instances of the cut-rule.

A *logical rule* in $\mathcal{L}$ is an inference rule that introduces a logical connective (e.g., $\wedge, \vee, \cdot, \backslash, /$) on the left or right of the denominator. It is said to have the *subformula property* if for all instances of the rule, all formulas appearing in the numerator are subformulas of the denominator. If $\mathcal{L}$ has logical rules with the subformula property and the structural rules do not increase complexity, then for any sequent s the set $s^{\leftarrow}$ is finite. It is easy to verify that $\mathbf{FL}$ (see Figure 1.1) and its extensions by simple rules have the subformula property. In [8], the following is proved:

Proposition 3.3.1 ([8])**.** $\mathbf{FL}$ and its extensions with simple rules that do not increase complexity have the FMP.

Here, the finite countermodel falsifying the provability of a sequent s is constructed from a residuated frame which encodes membership of $s^{\leftarrow}$, where the finiteness of $s^{\leftarrow}$ guarantees the finiteness of the resulting algebra.

We say that a simple equation $[\mathrm{R}] = (\mathbf{1_n}, \mathrm{R})$ is *completely linear* if the set of terms R is linear, i.e., for each $r \in \mathrm{R}$ and $i \in [1, k]$, $\#(r, x_i) \leq 1$. For instance, commutativity $(xy \leq$

[5] A *well partially-ordered set* is a poset $(A, \leq)$ that contains no infinite antichains, i.e., every infinite set $X \subseteq A$ contains a pair $x, y \in X$ such $x \leq y$.

$yx)$ is completely linear, integrality and (the linearization of) k-mingle are all completely linear,[6] and non-pre-knotted equations such as

$$xyz \leq xy \vee xz \vee yz \vee 1, \tag{3.2}$$

are completely linear. Analogously, we say a simple rule (R) is *completely linear* if the simple equation $[\mathrm{R}]$ is completely linear. E.g., the simple rule (R) obtained from Equation (3.2) *vis-à-vis* Equation (2.1) yields the following completely linear rule:

$$\frac{\Delta_1, \Gamma_1, \Gamma_2, \Delta_2 \Rightarrow \Pi \quad \Delta_1, \Gamma_1, \Gamma_3, \Delta_2 \Rightarrow \Pi \quad \Delta_1, \Gamma_2, \Gamma_3, \Delta_2 \Rightarrow \Pi \quad \Delta_1, \Delta_2 \Rightarrow \Pi}{\Delta_1, \Gamma_1, \Gamma_2, \Gamma_3, \Delta_2 \Rightarrow \Pi}\ (\mathrm{R}) \tag{3.3}$$

It is immediate the simple rule (R) from Equation (3.3) does not increase complexity of length, since each sequent in the numerator contains no more instances of a metavariable than those which occur in the denominator. This property holds for any completely linear simple rule by definition of each term of the numerator being linear and containing no metavariable not contained in the denominator (i.e., the rule is proper). Therefore, if Σ is a finite set of completely linear simple rules, then the proof searches in $\mathbf{FL} + \Sigma$ will be finitely branching and will not increase the complexity of length. Consequently, by Proposition 3.3.1, we obtain:

Theorem 3.3.2. $\mathbf{FL} + \Sigma$ has the FMP for any set finite Σ of completely linear simple rules.

We note that, as a consequence of [15], the decision problem for provability in $\mathbf{FL} + \Sigma$ is PSPACE-complete for any finite set Σ of completely linear simple rules. In fact, the results [15] entail that provability in $\mathbf{FL} + \Sigma$ is PSPACE-hard for any set Σ of simple rules, providing a fundamental lower-bound for complexity of any decision procedure.

[6] The equation $x^k \leq x$ is called *k-mingle*, and its linearization is given by $[\mathsf{K}^1_k]$, see Section 2.4.

3.4 Potent Commutative Varieties

Although the FMP guarantees provability is decidable for a given logic, the demonstration of this fact is inherently non-constructive and so the direct implementation of the decision algorithm is not feasible. Notably, while $\mathsf{CRL_c}$ has the FEP, Urquhart showed in [23] that the decision procedure for provability in $\mathbf{FL_{ec}}$ is not primitive recursive.[7]

However, we will show that potent substructural logics admit primitive recursive decision procedures. This demonstration is a natural generalization of the prototypical decidability proof for $\mathbf{FL_{ecw}}$ given in [11] by Gentzen.

3.4.1 Sequents in $\mathbf{FL_e}$ and the $*$ function. By the presence of the exchange rule,

$$\frac{\Delta_1, \Psi, \Gamma, \Delta_2 \Rightarrow \Pi}{\Delta_1, \Gamma, \Psi, \Delta_2 \Rightarrow \Pi}\ (\mathsf{e})$$

the antecedent of a sequent in $\mathbf{FL_e}$ may be represented by a different data-type, namely that of a *multiset* of formulas instead of a sequence of formulas. For a function $X : Fm \to \mathbb{N}$, we write $|X|_a := X(a)$ as the value (or *multiplicity*) of a formula $a \in Fm$ in X. By $[X]$ denote the Fm-*support* of X, where $[X] := \{a \in Fm : |X|_\alpha > 0\}$. If $[X]$ is finite, we say the function X is a *multiset*, and we typically view X as a finite unordered list of formulas from $[X]$, with possible repetitions, where each $a \in [X]$ occurs exactly $|X|_a$ many times. We define X, Y to be the addition of multisets X and Y, where $[X, Y] = [X] \cup [Y]$ and $|X, Y|_a = |X|_a + |Y|_a$ for each $a \in Fm$. Let $Mset$ denote the collection of all multisets. Note that $Mset$ forms a monoid with multiset addition as defined above.

Fix $n \in \mathbb{N}$. For a given formula $a \in Fm$, we define a^n to be the multiset with support $[a^n] = \{a\}$ such that $|a^n|_a = n$. Similarly, given a multiset X, we define X^n to be the multiset with support $[X^n] = [X]$ such that $|X^n|_b = n \cdot |X|_b$ for each $b \in Fm$.

[7] We will revisit and extend this construction in Section 4.5.

$$\frac{X \Rightarrow \Pi}{X, 1 \Rightarrow \Pi}\ (1l) \qquad \frac{X \Rightarrow}{X \Rightarrow 0}\ (0r)$$

$$\frac{X, \alpha, \beta \Rightarrow \Pi}{X, \alpha \cdot \beta \Rightarrow \Pi}\ (\cdot l) \qquad \frac{X \Rightarrow \alpha \quad Y \Rightarrow \beta}{X, Y \Rightarrow \alpha \cdot \beta}\ (\cdot r)$$

$$\frac{X \Rightarrow \alpha \quad Y, \beta \Rightarrow \Pi}{X, Y, \alpha \to \beta \Rightarrow \Pi}\ (\to l) \quad \frac{X, \alpha \Rightarrow \beta}{X \Rightarrow \alpha \to \beta}\ (\to r)$$

$$\frac{X, \alpha \Rightarrow \Pi \quad X, \beta \Rightarrow \Pi}{X, \alpha \vee \beta \Rightarrow \Pi}\ (\vee l) \quad \frac{X \Rightarrow \beta}{X \Rightarrow \alpha \vee \beta}\ (\vee r_1) \quad \frac{X \Rightarrow \alpha}{X \Rightarrow \alpha \vee \beta}\ (\vee r_2)$$

$$\frac{X \Rightarrow \alpha \quad X \Rightarrow \beta}{X \Rightarrow \alpha \wedge \beta}\ (\wedge r) \quad \frac{X, \beta \Rightarrow \Pi}{X, \alpha \wedge \beta \Rightarrow \Pi}\ (\wedge l_1) \quad \frac{X, \alpha \Rightarrow \Pi}{X, \alpha \vee \beta \Rightarrow \Pi}\ (\wedge l_2)$$

Figure 3.1: Logical rules of $\mathbf{FL_e}$, where $\alpha, \beta \in \mathrm{FmV}$, Π either empty or in FmV, $0, 1 \in \mathrm{ConV}$, and $X, Y, Z \in \mathrm{MsV} \setminus (\mathrm{FmV} \cup \mathrm{ConV})$.

Recall that a *sequent* in $\mathbf{FL_e}$ is an expression of the form $X \Rightarrow \Pi$, for some $X \in Mset$ and Π is either a formula or the emptyset. For our purposes, it will be important to distinguish between a *metasequent* and an *instance of a metasequent* (which is a sequent).

We define four classes of metavariables. Let $\mathrm{FmV} = \{\alpha, \beta, \ldots\}$ denote *formula-type*, $\mathrm{ConV} = \{1, 0\} \cup \{\alpha \star \beta : \alpha, \beta \in \mathrm{FmV}, \star \in \{\vee, \wedge, \cdot, \to\}\}$ denote *connective-formula-type*, MsV denote *multiset-type* with $\mathrm{FmV} \cup \mathrm{ConV} \subset \mathrm{MsV}$, and $\mathrm{MsV}^* = \{X^* : X \in \mathrm{MsV}\}$ denote $*$*-multiset-type*. A *metasequent* s is an expression of the form $V_1, \ldots, V_k \Rightarrow V_{k+1}$, where $\{V_i\}_{i=1}^k \subset \mathrm{MsV} \cup \mathrm{MsV}^*$ and V_{k+1} is either in $\mathrm{FmV} \cup \mathrm{ConV}$ or is the empty word.

A *valuation* $\nu : \mathrm{FmV} \to Fm$ is a function assigning each $\alpha \in \mathrm{FmV}$ to a formula $\nu(\alpha) \in Fm$ and the empty word to the emptyset. We will denote $\nu(\alpha)$ by $\overline{\alpha}$ if the valuation is understood in context. Given such a valuation, we may extend it to a function $\nu : \mathrm{MsV} \cup \mathrm{MsV}^* \to Fm \cup Mset$ via $\overline{\alpha \star \beta} = \overline{\alpha} \star \overline{\beta}$ for any $\alpha \star \beta \in \mathrm{ConV}$, $\overline{X} \in Mset$ for any

$X \in \mathrm{MsV} \setminus (\mathrm{FmV} \cup \mathrm{ConV})$, and

$$\overline{V^*} = \begin{cases} \overline{V}, & \text{if } V \in \mathrm{FmV} \cup \mathrm{ConV} \\ * \circ \overline{V}, & \text{otherwise} \end{cases},$$

for $V^* \in \mathrm{MsV}^*$, where $* : \mathbb{N} \to \mathbb{N}$ is a function that will be defined below. Note that we may view the range of ν as a subset of $Mset$ by the embedding $a \mapsto a^1$, for any $a \in Fm$. Given a valuation, we define an *instance of a metasequent* $s = V_1, \ldots, V_k \Rightarrow V_{k+1}$, denoted by $\nu(s)$ (or by $\overline{s}$ if clear by context), to be the sequent $\overline{V_1}, \ldots, \overline{V_k} \Rightarrow \overline{V}_{k+1}$. We let V^n be shorthand for the n successive occurrences $V, \ldots, V$, for any $V \in \mathrm{MsV} \cup \mathrm{MsV}^*$. If $n = 0$, then a metasequent such as $Y, X^n \Rightarrow \Pi$ denotes $Y \Rightarrow \Pi$.

Now, for $n, m > 0$, we define $*_n^m : \mathbb{N} \to [0, n+m)$ via

$$*_n^m(x) = \begin{cases} x & \text{where } x < n+m, \text{otherwise;} \\ x - q_x m & \text{where } q_x \in \mathbb{Z}^+ \text{ and } x - q_x m \in [n, n+m) \end{cases}$$

When the values n, m are understood in context, we will simply write $* := *_n^m$.

Lemma 3.4.1. For all $a, b \in \mathbb{N}$, $*(a+b) = *(*(a) + *(b))$

Proof. Note that for each $x \in \mathbb{N}$, $*(x) = x - q_x m$ for some $q_x \in \mathbb{N}$ and $*(x) < n+m$. It is easy to see that $*$ is idempotent and non-increasing. Furthermore, $*(x) = *(x - qm)$ for any $0 \leq q \leq q_x$. Note that if $*(a) + *(b) < n$ then $*(a), *(b) < n$, and hence $*(a) = a$ and $*(b) = b$ and we are done. So we may assume $*(a) + *(b) \geq n$. Since $*$ is non-increasing it follows $a + b \geq n$, and so

$$*(a+b) = (a+b) - q_{a+b} m \in [n, n+m).$$

But $*(a) + *(b) = (a + b) - (q_a + q_b)m \in [n, 2n + 2m)$. Since q_{a+b} is the least such number k such that $(a + b) - km \in [n, n + m)$, we have that $q_a + q_b \leq q_{a+b}$. Hence $*(*(a) + *(b)) = *(a + b)$. □

Let $[\mathrm{R}] = (\mathbf{1_k}, \mathrm{R})$ be a simple equation. For each $r \in \mathrm{R}$, define

$$r^* := \prod_{i=1}^{k} x_i^{*(n_i)},$$

where $n_i = \#(r, x_i)$ for each $i = 1, \ldots, k$, and let $\mathrm{R}^* = \{r^* : r \in \mathrm{R}\}$. We define the simple equation $[\mathrm{R}^*] := (\mathbf{1_k}, \mathrm{R}^*)$.

Lemma 3.4.2. Let Σ be a set of simple equations. Then $\mathsf{CRL} + (x^n = x^{n+m}) + \Sigma$ is equivalent to $\mathsf{CRL} + (x^n = x^{n+m}) + \Sigma^*$, where $\Sigma^* := \{[\mathrm{R}^*] : [\mathrm{R}] \in \Sigma\}$.

Proof. Let $[\mathrm{R}] = (\mathbf{1_k}, \mathrm{R}) \in \Sigma$. For each $r \in \mathrm{R}$, define $r(i) = \#(r, x_i)$ for each $i = 1, \ldots, k$. So $\mathsf{CRL} \models r = \prod_{i=1}^{k} x^{r(i)}$. Since $r^* := \prod_{i=1}^{k} x^{r(i)}$, we obtain $\mathsf{CRL} + (x^n = x^{n+m}) \models r = r^*$. Therefore it follows that $\mathsf{CRL} + (x^n = x^{n+m}) + [\mathrm{R}] \models [\mathrm{R}^*]$ and $\mathsf{CRL} + (x^n = x^{n+m}) + [\mathrm{R}^*] \models [\mathrm{R}]$. □

3.4.2 $*$-sequents and inference rules. Given a metasequent s given by $\Upsilon \Rightarrow \Psi$, we write s^* to denote the metasequent $\Upsilon^* \Rightarrow \Psi$, where $\Upsilon^* := V_1^*, \ldots, V_k^*$ if $\Upsilon = V_1, \ldots, V_k$. For a sequent t given by $X \Rightarrow \Pi$, by ${}^*t$ we denote the sequent $*X \Rightarrow \Pi$, where $*X := * \circ X$. Given a set of metasequents Γ and a set of sequents Δ, we write $\Gamma^* := \{s^* : s \in \Gamma\}$ and ${}^*\Delta := \{{}^*t : t \in \Delta\}$. For an inference rule (r) given by $\frac{\Gamma}{s}(r)$, we define the rule $(r)^*$ via $\frac{\Gamma^*}{s^*}(r)^*$. By the definition of valuations, it is easy to check that for every inference rule (r) of $\mathbf{FL_e}$, an instance of $(r)^*$ is an instance of (r): E.g.,

$$\frac{X^* \Rightarrow \alpha \quad Y^*, \beta^* \Rightarrow \gamma}{X^*, Y^*, \alpha^* \to \beta^* \Rightarrow \gamma}\ (\to l)^*,$$

and for a valuation ν, $\overline{V_1^*} = \overline{V_1}$ if $V_1 \in \mathrm{FmV}$ and $\overline{V_2^*} = *\overline{V_2}$ if $V_2 \in \mathrm{MsV}$, and so both $(\nu, (\to l)^*)$ and ${}^*(\nu, (\to l))$ are equivalent to the following instance:

$$\frac{*\overline{X} \Rightarrow \overline{\alpha} \quad *\overline{Y}, \overline{\beta} \Rightarrow \overline{\gamma}}{*\overline{X}, *\overline{Y}, \overline{\alpha} \to \overline{\beta} \Rightarrow \overline{\gamma}} \ .$$

Furthermore, any metasequent $\Upsilon \Rightarrow \Psi$ that appears in an inference rule (r) for $\mathbf{FL_e}$ (see Figure 3.1) is one of three possible forms: (i) $X \Rightarrow \Psi$, (ii) $X, Y \Rightarrow \Psi$, or (iii) $X, Y, \alpha \Rightarrow \Psi$, where $X, Y \in \mathrm{MsV}$, $\alpha \in \mathrm{ConV}$, and Ψ is either empty or in $\mathrm{FmV} \cup \mathrm{ConV}$. Hence for any formula $a \in Fm$ and inference rule (r) for $\mathbf{FL_e}$ with conclusion $\Upsilon \Rightarrow \Psi$, we have that $|\overline{\Upsilon^*}|_a \leq 2(n+m) - 1$, since (iii) is the most complicated form of Υ and

$$|\overline{\Upsilon^*}|_a \leq |\overline{X^*}, \overline{Y^*}, \overline{\alpha}|_a = |*\overline{X}|_a + |*\overline{Y}|_a + |\overline{\alpha}|_a \leq 2 \cdot (n+m-1) + 1, \tag{3.4}$$

If $[\mathrm{R}] = (\mathbf{1_k}, \mathrm{R})$ is a simple equation, then the simple rule (R) is $\mathbf{FL_e}$-equivalent to the following inference rule

$$\frac{\{Y, X_1^{r(1)}, \ldots, X_k^{r(k)} \Rightarrow \Pi\}_{r \in \mathrm{R}}}{Y, X_1, \ldots, X_k \Rightarrow \Pi} \ (\mathrm{R})_{\mathsf{e}},$$

where $r(i) := \#(r, x_i)$, for each $r \in \mathrm{R}$ and $1 \leq i \leq k$. Similarly, by the definition of valuations $(\mathrm{R})^*_{\mathsf{e}}$ is an instance of $(\mathrm{R})_{\mathsf{e}}$:

$$\frac{\{Y^*, (X_1^*)^{r(1)}, \ldots, (X_k^*)^{r(k)} \Rightarrow \Pi\}_{r \in \mathrm{R}}}{Y^*, X_1^*, \ldots, X_k^* \Rightarrow \Pi} \ (\mathrm{R})_{\mathsf{e}}.$$

Thus, for any metasequent $\Upsilon \Rightarrow \Psi$ which appears in $(\mathrm{R})_{\mathsf{e}}$ and $a \in Fm$, it follows that

$$\begin{aligned} \left|\overline{\Upsilon^*}\right|_a = \left|\overline{Y^*}, \overline{(X_1^*)^{r(1)}}, \ldots, \overline{(X_k^*)^{r(k)}}\right|_a &= \left|\overline{Y^*}\right|_a + \textstyle\sum_{i=1}^k |X_i^*|_a r(i) \\ &\leq (n+m-1)\left(1 + \textstyle\sum_{i=1}^k r(i)\right), \end{aligned} \tag{3.5}$$

where $r \in \mathrm{R} \cup \{\mathbf{1_k}\}$. Hence $\left|\overline{\Upsilon^*}\right|_a \leq (n+m-1)\left(1+\sum_{i=1}^{k} r(i)\right)$ for any $a \in Fm$ and metasequent $\Upsilon \Rightarrow \Psi$ of $(\mathrm{R})_{\mathsf{e}}$, where $M_{[\mathrm{R}]} := \max\{\sum_{i=1}^{k} r(i) : r \in \mathrm{R} \cup \{\mathbf{1_k}\}\}$.

3.4.3 Reduced proofs for potent varieties. Let $n, m > 0$ and set $* := *_n^m$. Let Σ be a finite set of simple equations and define $M_\Sigma := \max\{M_{[\mathrm{R}]} : [r] \in \mathrm{R}\}$. Henceforth, we define

$$\mathcal{L} := \mathbf{FL}_{\mathsf{e}} + (\mathsf{K}_n^{n+m}) + (\mathsf{K}_{n+m}^{n}) + \Sigma_{\mathsf{e}}.$$

We note that $\mathcal{L}$ is cut-admissible by [8] since it is an extension of $\mathbf{FL}$ by simple rules. By definition (K_n^{n+m}) and (K_{n+m}^{n}),[8] $\mathcal{L}$ satisfies the following inference rules:

$$\frac{Y, \alpha^{n+m} \Rightarrow \Pi}{Y, \alpha^{n} \Rightarrow \Pi}\,(\uparrow^m) \qquad \& \qquad \frac{Y, \alpha^{n} \Rightarrow \Pi}{Y, \alpha^{n+m} \Rightarrow \Pi}\,(\downarrow^m),$$

where $Y \in \mathrm{MsV}$ and $\alpha \in FV$.

Lemma 3.4.3. Let t be any sequent. Then t is provable in $\mathcal{L}$ if and only if ${}^*t$ is provable in $\mathcal{L}$. In particular, ${}^*t \vdash_{\mathcal{L}} t$ and $t \vdash_{\mathcal{L}} {}^*t$.

Proof. Suppose $\vdash_{\mathcal{L}} {}^*t$. Let t be given by $X \Rightarrow \Pi$. For each $a \in [X]$ we will apply $(\downarrow^m)$ sequentially to obtain the proper multiplicity of a in X. If $*|X|_a = |X|_a$, we need do nothing. Otherwise, $*|X|_a = |X|_a - q_a m \in [n, n+m)$, where $q_a > 0$. First, we repeatedly apply the exchange rule (e) to obtain the sequent $*X^-, a^{|X|_a - q_a m} \Rightarrow \Pi$, where X^- is the multiset such that $|X^-|_a = 0$ but $|X^-|_b = |X|_b$ for all $b \in Fm \setminus \{a\}$. Then we may apply rule $(\downarrow^m)$ exactly q_a many times to the formula a.[9] Once this has been completed for each $a \in [X]$, by applying the exchange rule repeatedly we will have derived t. See Figure 3.2.

[8] See Section 2.4.

[9] By this we mean that we apply instances $(\downarrow^m)$ to a sequentially such that $t = \nu_1(s_\pi)$ and $\nu_i(s_c) = \nu_{i+1}(s_\pi)$ for $i = 1, \ldots, q_A - 1$, where s_π and s_c are the premise and conclusion of $(\downarrow^m)$, respectively.

Similarly, suppose $\vdash_{\mathcal{L}} t$. For each $a \in [X]$ we will apply $(\uparrow^m)$ sequentially to obtain the proper multiplicity of a in $*X$. If $|X|_a = *|X|_a$, we need do nothing. Otherwise, $|X|_a \geq n+m$, so $*|X|_a = |X|_a - q_a m \in [n, n+m)$, where $q_a > 0$. Similarly to the above, we begin by applying the exchange rule to put the sequent in the proper form, and then we apply rule $(\uparrow^m)$ exactly q_a many times to the formula a. Once this has been completed for each $a \in [X]$, we will have derived ${}^*t$ by possible further applications of exchange. See Figure 3.2.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{{}^*t}{\vdots}\,(\mathsf{e})}{*X^-, a^{|X|_a - q_a m} \Rightarrow \Pi}\,(\mathsf{e})}{*X^-, a^{|X|_a - q_a m + m} \Rightarrow \Pi}\,(\downarrow^m)}{\vdots}\,(\downarrow^m)}{*X^-, a^{|X|_a - m} \Rightarrow \Pi}\,(\downarrow^m)}{*X^-, a^{|X|_a} \Rightarrow \Pi}\,(\downarrow^m) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{t}{\vdots}\,(\mathsf{e})}{X^-, a^{|X|_a + q_a m} \Rightarrow \Pi}\,(\mathsf{e})}{X^-, a^{*|X|_a + q_a m - m} \Rightarrow \Pi}\,(\uparrow^m)}{\vdots}\,(\uparrow^m)}{X^-, a^{*|X|_a + m} \Rightarrow \Pi}\,(\uparrow^m)}{X^-, a^{*|X|_a} \Rightarrow \Pi}\,(\uparrow^m)$$

Figure 3.2: Proof heuristic for reduced sequents. Here, X^- is the multiset such that $|X^-|_a = 0$ but $|X^-|_b = |X|_b$ for all $b \in Fm \setminus \{a\}$.

□

By Lemma 3.4.2, we may assume $\Sigma = \Sigma^*$. Note that this implies $M_\Sigma \leq (n+m-1)m_\Sigma$, where $m_\Sigma := \max\{x \in \mathbb{Z}^+ : (\mathbf{1}_{\mathbf{x}}, \mathrm{R}) \in \Sigma\}$. Define $M_{\mathcal{L}} := \max\{3, M_\Sigma + 1\}$, which represents the maximum number of metavariables that can can appear in a metasequent of an inference rule from $\mathcal{L}$. By Equations (3.4) and (3.5), it follows that for any metasequent $\Upsilon \Rightarrow \Psi$ appearing in an inference rule in $\mathcal{L}$, valuation ν, and formula $a \in Fm$.

$$|\overline{\Upsilon^*}|_a \leq M_{\mathcal{L}}(n+m-1). \tag{3.6}$$

Let Δ be a (possibly empty) set of sequents. We say a proof of a sequent t from Δ in $\mathcal{L}$ is *k-reduced*, denoted $\Delta \vdash^k_{\mathcal{L}} t$, if there is a proof from Δ to t in which every sequent $X \Rightarrow \Pi$ in the proof-tree of t is such that $|X|_a \leq k$ for each formula $a \in Fm$.

Lemma 3.4.4. Set $k := M_{\mathcal{L}}(n+m-1)$. Let s be a metasequent from an inference rule in $\mathcal{L}$. Then for any valuation ν,

1. ${}^*\overline{s} \vdash^k_{\mathcal{L}} \overline{s^*}$.
2. $\overline{s^*} \vdash^k_{\mathcal{L}} {}^*\overline{s}$.

Proof. Let s be a metasequent as above, given by $X_1, \ldots, X_N \Rightarrow \Pi$, and ν a valuation.[10] Note that $N \leq M_{\mathcal{L}}$ by definition of $M_{\mathcal{L}}$.

(1) For a formula $a \in Fm$, observe that $*|\overline{X_1}, \ldots, \overline{X_N}|_a = *(\sum_{i=1}^N |\overline{X_i}|_a)$. If

$$* \left(\sum_{i=1}^{N} |\overline{X_i}|_a \right) < n+m,$$

then by definition of $*$ this implies $* \left(\sum_{i=1}^N |\overline{X_i}|_a \right) = \sum_{i=1}^N |\overline{X_i}|_a$, so $|\overline{X_i}|_a < n+m$ and thus $*|\overline{X_i}|_a = |\overline{X_i}|_a$ for each $i \leq N$. It then follows that $*|\overline{X_1}, \ldots, \overline{X_N}|_a = |*\overline{X_1}, \ldots, *\overline{X_N}|_a$ and we need do nothing. Otherwise,

$$* \left(\sum_{i=1}^{N} |\overline{X_i}|_a \right) = \sum_{i=1}^{N} |\overline{X_i}|_a - q_a m \in [n, n+m),$$

for $q_a > 0$. By Lemma 3.4.1, we have that

$$* \left(\sum_{i=1}^{N} *|\overline{X_i}|_a \right) = * \left(\sum_{i=1}^{N} |\overline{X_i}|_a \right).$$

[10] If any of the X_i's are in $\mathrm{FmV} \cup \mathrm{ConV}$, view them as multisets via the embedding described in Section 3.4.1.

Apply rule $(\downarrow^m)$ exactly q_a many times to obtain precisely $\sum_{i=1}^{N} *|\overline{X}_i|_a$ occurrences of a.[11] Once this has been completed for each formula a, we will have derived $*\overline{X}_1, \ldots, *\overline{X}_N \Rightarrow \overline{\Pi}$, which is exactly $\overline{s^*}$. Furthermore, for any formula $b \in Fm$, since we only used the rule $(\downarrow^m)$, the most occurrences of b that appear in the antecedent of a sequent in this proof occur after the final application of $(\downarrow^m)$, so by Equation (3.6)

$$|*\overline{X_1}, \ldots, *\overline{X_N}|_b = \sum_{i=1}^{n} *|\overline{X_i}|_b \leq \sum_{i=1}^{M_{\mathcal{L}}} n + m - 1 = M_{\mathcal{L}}(n + m - 1) = k.$$

Hence ${}^*\overline{s} \vdash_{\mathcal{L}}^{k} \overline{s^*}$.

(2) For a formula a, observe that $|*\overline{X_1}, \ldots, *\overline{X_N}|_a = \sum_{i=1}^{N} *|\overline{X}_i|_a$. If

$$\sum_{i=1}^{N} *|\overline{X}_i|_a = * \left(\sum_{i=1}^{N} |\overline{X}_i|_a \right),$$

then $|*\overline{X}_1, \ldots, *\overline{X}_N|_a = *|\overline{X}_1, \ldots, \overline{X}_N|_a$ and we need do nothing. Otherwise,

$$\sum_{i=1}^{N} *|\overline{X_i}|_a = n + r_a + q_a m,$$

where $0 \leq r_a < m$ and $q_a > 0$. Apply rule $(\downarrow^m)$ exactly q_a many times to obtain precisely $*(\sum_{i=1}^{N} *|\overline{X_i}|_a)$ occurrences of a. But by Lemma 3.4.1,

$$* \left(\sum_{i=1}^{N} *|\overline{X_i}|_a \right) = * \left(\sum_{i=1}^{N} |\overline{X_i}|_a \right).$$

Once this has been completed for each formula we will have obtained $*(\overline{X_1}, \ldots, \overline{X_N}) \Rightarrow \overline{\Pi}$, which is exactly ${}^*\overline{s}$. Since we only used the rule $(\downarrow^m)$, it follows that $\overline{s^*} \vdash_{\mathcal{L}}^{k} {}^*\overline{s}$. □

Theorem 3.4.5. Let t be a sequent. Then $\vdash_{\mathcal{L}} t$ iff $\vdash_{\mathcal{L}}^{k} {}^*t$.

[11] As was shown in Lemma 3.4.3 and Figure 3.2.

Proof. The ($\Leftarrow$) direction follows from Lemma 3.4.3.

($\Rightarrow$) Since t is provable in $\mathcal{L}$, by [8] there is a cut-free proof of t in $\mathcal{L}$. We proceed by induction on the height n of the cut-free proof of t. If $n = 1$ then $(r) \in \{(\mathbf{init}), (1r), (0l)\}$, and it must be that ${}^*t = t$, and we are done by our assumption.

So suppose the proof of $\overline{s}$ has height $n > 1$. By definition, t is labeled by $(\nu, (r))$, and $t = \overline{s_0} := \nu(s_0)$ where s_0 is the conclusion of (r), an inference rule in $\mathcal{L}$. Then

$$\frac{\overline{\Gamma}}{\overline{s_0}}\ (\overline{r}),$$

where Γ is the set premises of (r) and $\overline{\Gamma} = \{\overline{s} : \overline{s} = v(s)\ \&\ s \in \Gamma\}$ is the set of sequents that are the immediate children of $\overline{s_0}$ in the proof-tree. Each sequent in Γ therefore has a proof-tree of height strictly less than n and is the conclusion of some inference rule. Hence by the inductive hypothesis, $\vdash^k_{\mathcal{L}} {}^*\overline{s}$ for each $\overline{s} \in \overline{\Gamma}$. Given $\overline{s} \in \overline{\Gamma}$, we have ${}^*\overline{s} \vdash^k_{\mathcal{L}} \overline{s^*}$ by Lemma 3.4.4(1). Thus we obtained $\overline{s_0^*}$ via

$$\frac{\overline{\Gamma^*}}{\overline{s_0^*}}\ \nu'(r),$$

for a valuation ν' such that $\{\nu'(s) : s \in \Gamma\} = \overline{\Gamma^*}$ and $\nu'(s_0) = \overline{s_0^*}$. By Lemma 3.4.4(2) we obtain $\overline{s_0^*} \vdash^k_{\mathcal{L}} {}^*\overline{s_0}$. Therefore $\vdash^k_{\mathcal{L}} {}^*t$. □

3.4.4 The decision procedure. In this way, for a given sequent s, a proof-search for s need only consider k-reduced proofs of ${}^*s$. The complexity of the proof-search for ${}^*s$ can be crudely bounded above as follows:

Suppose ${}^*s$ is given by $*X \Rightarrow \Pi$, and let $N := |\mathrm{SubFm}(*X, \Pi)|$ denote the total number of subformulas from formulas appearing in $*X$ and Π. Let T denote the total number of k-reduced sequents constructed from $\mathrm{SubFm}(*X, \Pi)$. Since each of the N-many formulas in $\mathrm{SubFm}(*X, \Pi)$ can appear in the antecedent at most k-many times, there

are $(k+1)^N$ many possible antecedents. Since there are N-many possible formulas that can appear as the consequent, the total number of k-reduced sequents is $T = N(k+1)^N$. Let $({}^*s)^{\leftarrow k}$ denote the set of all possible k-reduced sequents in ${}^*s^{\leftarrow}$.[12] Since $\mathcal{L}$ satisfies the subformula property, $|({}^*s)^{\leftarrow k}| \leq T$. The *height* of a proof-attempt is the total number of sequents along the line of the tallest branch. The maximum height of a proof-attempt cannot exceed the total number of k-reduced sequents T. Indeed, we may omit proof-attempts that contain duplicate occurrences of a sequent along the line of a single branch. Therefore, an exhaustive proof-search of ${}^*s$ need only consider proof-attempts whose height does not exceed T. We provide a bound P_s for the total number of proof-attempts of height T, which will be doubly-exponential with respect to the number of subformula N.

We first find a bound for the maximum number $R_{\mathcal{L}}$ of instances of rules from $\mathcal{L}$ in which a sequent in $({}^*s)^{\leftarrow k}$ can be the conclusion. Let R_{log} be the total number of instances of logical rules and R_{str} be the total number of structural rules, for which a sequent in $({}^*s)^{\leftarrow k}$ can appear as their conclusion. It is then clear that $R_{\mathcal{L}} \leq R_{log} + R_{str}$. Since logical rules are only applicable to the formulas that appear in a sequent, and the structural rules can only be applied to the appropriate partitioning of the multiset in a sequent's antecedent, the values R_{log} and R_{str} can be bounded by inspecting sequents that contain the maximum number of formulas. By N_w we denote the maximum number or formulas that can appear in a sequent t from $({}^*s)^{\leftarrow k}$. Every formula appearing in t is a subformula of ${}^*s$ and can appear at most k-many times since every element of $({}^*s)^{\leftarrow k}$ is k-reduced. Thus the sequent t can have at most $k \cdot N$ many formulas in its antecedent, a single formula as its consequent, and therefore the total number of formulas in t no larger than $N_w = kN + 1$.

Now, logical rules can only be applied to formulas with the appropriate outermost connective. For any formula α, either α is a constant in $\{0, 1\}$, a propositional variable,

[12] Recall, $t^{\leftarrow}$ is the least set of sequents that can appear in any proof-search with sequent t as the root. See Section 3.3.

or $\alpha = \beta \star \gamma$ for formulas β, γ and outermost connective in $\star \in \{\vee, \wedge, \cdot, \rightarrow\}$. The only (non-structural) rule applicable to a propositional variable is an instance of $(\mathbf{init})$. Recall Figure 3.1. For α a constant 0 or 1 there are at most two unique instances from the rules $(\mathbf{init}), (0r)$ and $(1l)$ applicable to α. If $\alpha = \beta \star \gamma$ then there are at most two unique instances of rules for $\star \in \{\vee, \wedge\}$. Namely, the pairs $(\vee r_1), (\vee r_2)$ if $\alpha = \beta \vee \gamma$ is the consequent, and $(\wedge l_1), (\wedge l_2)$ if $\alpha = \beta \wedge \gamma$ in the antecedent. However, for $\star \in \{\cdot, \rightarrow\}$ there are at most 2^{N_w} possible applications of a rule. Namely, for $(\cdot r)$ if $\alpha = \beta \cdot \gamma$ is the consequent, or for $(\rightarrow l)$ if $\alpha = \beta \rightarrow \gamma$ in the antecedent. Indeed, for $(\cdot r)$ as an example, since there are no more than N_w-many formulas in a sequent $t \in ({}^*s)^{\leftarrow k}$, there are at most 2^{N_w} many ways to write $t = X_w, Y_w \Rightarrow \alpha$, so each representation of X_w, Y_w as the antecedent corresponds to an instance of $(\cdot r)$. Hence, for a given formula α, there are at most 2^{N_w}-many applications of a logical rule applicable to a sequent containing α in all cases. Therefore there are at most $R_{log} = N_w \cdot 2^{N_w}$ instances of structural applicable to any sequent in $({}^*s)^{\leftarrow k}$.

We now consider how many possible instances of structural rules from $\mathcal{L} = \mathbf{FL_e} + (\mathsf{K}_n^{n+m}) + (\mathsf{K}_{n+m}^n) + \Sigma_\mathsf{e}$ are applicable to a given sequent. Define $M_\mathcal{L}$ to be the maximum number of multiset variables which appear in the conclusion of a structural rule from the set $\Gamma := \{(\mathsf{K}_n^{n+m}), (\mathsf{K}_{n+m}^n)\} \cup \Sigma_\mathsf{e}$. E.g., if $\Sigma_\mathsf{e} = \emptyset$ then $M_\mathcal{L} = (n+m)+1$ since (K_{n+m}^n) contains $(n+m)+1$ many multiset variables in its conclusion. Since the cut rule is admissible, we need only consider structural rules from the set Γ. Such structural rules are applicable to a sequent only when a partition of its antecedent into (at most) $M_\mathcal{L}$-many multisets is chosen. Since there are no more than N_w formulas that appear in the antecedent of $t \in ({}^*s)^{\leftarrow k}$, and each formula can be contained in one of the $M_\mathcal{L}$-many multisets, there are at most $M_\mathcal{L}^{N_w}$-many ways to partition the antecedent of t into $M_\mathcal{L}$-many multisets. Thus for any given rule in Γ, there are at most $M_\mathcal{L}^{N_w}$-many instances applicable to t. Hence there are at most $R_{str} = |\Gamma| \cdot M_\mathcal{L}^{N_w}$ many instances of structural rules applicable to any sequent in $({}^*s)^{\leftarrow k}$.

In total, there are no more than $R_{log} + R_{str}$ many instances of rules in $\mathcal{L}$ that are applicable to any given sequent in $({}^*s)^{\leftarrow k}$. For a sequent $t \in ({}^*s)^{\leftarrow k}$ and number $h \geq 0$, by $P_{\#}(t, h)$ we denote the set of all proof-attempts with root t of height h. In this way, the maximum number of proof-attempts for ${}^*s$ of height T is given by $P_s := |P_{\#}({}^*s, T)|$. By the observations above, since there are at most $R_{\mathcal{L}}$-many rule instances in $\mathcal{L}$ where t is the conclusion, the value $|P_{\#}(t, 1)| \leq R_{\mathcal{L}}$. Note that every proof-attempt in $P_{\#}(t, 1)$ is of the form

$$\frac{t_1 \quad \cdots \quad t_B}{t} \ ,$$

with $B \leq B_{\mathcal{L}}$, where $B_{\mathcal{L}}$ is the maximum number of sequents that appear as the premises from rules in $\mathcal{L}$. That is, each proof-attempt in $P_{\#}(t, 1)$ has at most $B_{\mathcal{L}}$-many branches.

Now, every attempt in $P_{\#}(t, h+1)$ is a result of applying instances of rules from $\mathcal{L}$ to the leaves of an attempt in $P_{\#}(t, h)$ that are not axioms. Note that, if there is a proof-attempt in $P_{\#}(t, h)$ in which all leaves are axioms (i.e., empty leaves obtained by $(\mathbf{init})$), then t is provable and we *halt* the proof-search. Otherwise,

$$\begin{aligned} |P_{\#}(t, h+1)| \ &= \ (\#\text{ of proof-attempts in } P_{\#}(t, h)) \\ &\quad \times (\#\text{ of leaves in an attempt from } P_{\#}(t, h)) \\ &\quad \times (\#\text{ of rules applicable to a non-axiom leaf}) \\ &\leq \ |P_{\#}(t, h)| \cdot B_{\mathcal{L}}^{h} \cdot R_{\mathcal{L}} \end{aligned}$$

Hence, for each $h \geq 1$, $|P_{\#}(t, h)| \leq R_{\mathcal{L}}^{h} \cdot B_{\mathcal{L}}^{\triangle(h)}$, where $\triangle(h) := \sum_{i=1}^{h} i = \frac{h(h+1)}{2}$ is the h-th triangular number. Note that $\triangle(h) \leq h^2$ for all $h \geq 1$. Therefore, an exhaustive proof

search for ${}^*s$ need only check no more than

$$\begin{aligned} P_s &= |P_{\#}({}^*s, T)| \\ &\leq R_{\mathcal{L}}^T \cdot B_L^{\triangle(T)} \ , \\ &\leq R_{\mathcal{L}}^T \cdot B_L^{T^2} \end{aligned}$$

k-reduced proof attempts. That is, if this process has not halted in $R_{\mathcal{L}}^T \cdot B_L^{T^2}$ many proof-attempts, then ${}^*s$ is not provable. We recall the value $R_{\mathcal{L}} := R_{log} + R_{str} = N_w \cdot 2^{N_w} + |\Gamma| \cdot M^{N_w}$ is a bound for the maximum number of rule-instances applicable to a sequent in $({}^*s)^{\leftarrow k}$, where $N_w = kN + 1$ is the length a worst-case sequent, $T = N(k+1)^N$ is the total number of k-reduced sequents from the subformulas, and N is the total number of subformulas that appear in ${}^*s$. Since the values k, $B_{\mathcal{L}}$, $|\Gamma|$, and $M_{\mathcal{L}}$ are fixed for $\mathcal{L}$, we see that P_s as a function of sequents s is double-exponential in the number of subformulas N occurring in s. That is, in big-O notation,

$$P_s \leq O\left(\left(N_w \cdot 2^{N_w} + |\Gamma| \cdot M_{\mathcal{L}}^{N_w}\right)^{N(k+1)^N} \cdot B_L^{N^2(k+1)^{2N}}\right) = O\left(N^{N^N}\right) = O\left(2^{2^N}\right).$$

Theorem 3.4.6. $\mathcal{L}$ has a primitive recursive decision procedure. In particular, there is a decision procedure for $\mathcal{L}$ that is, at worst, double-exponential in the number of of subformula that appear in a given sequent.

Corollary 3.4.7. Let Σ be a finite potent set of simple equations. Then the equational, quasi-equational, and universal theories of $\mathsf{CRL} + \Sigma$ are decidable and admit a primitive recursive decision procedure.

While this proof-search is still computationally impractical, it demonstrates the possibility to cut down the complexity, unlike that for $\mathbf{FL}_{\mathsf{ec}}$. For instance, in [7] it is proved that the decision problem for extensions of $\mathbf{FL}_{\mathsf{ew}}$ by expansive knotted rules is in EXPTIME.

Chapter 4: Algebraic Machines and Complexity Lower-bounds

We now begin our investigation of complexity lower-bounds for the quasi-equational (and equational) theories for varieties of residuated lattices, specifically those which are defined by ISR-equations.

The fundamental decision problem we utilize is grounded in the halting problem for a class of abstract mathematical machines known as *counter machines*. The simplicity of the language for counter machines makes them distinctly well-suited for representations as ordered monoids or semirings. From this point of view, the key observations in [17, 14] are that *acceptance* of a machine (i.e., whether, for a given input, the machine halts) can be faithfully simulated as the satisfiability of a particular quasi-equation in a variety of residuated lattices. In fact, due to the fixed structure of a given machine, this correspondence relates to the complexity of the *word problem* for these algebraic structures. Inspired by [14], we use a residuated frames construction to prove the completeness of this result, while the soundness is easily achieved since residuated lattice have semiring reducts. In such a way, the varieties RL and CRL have been shown to have undecidable word problems. The first three sections of this chapter serve as the theoretical foundation in which we expand these results in Chapter 5. However, using the machinery developed in the first three sections, the final section is devoted to describing how [23] can be naturally extended to a broader class of simple equations. We note that the techniques developed in this section are not necessary for establishing the results in Chapter 5.

As shown in Chapter 2, residuated frames are also uniquely well-suited for the construction of residuated lattices that satisfy specific simple equations. In particular, we are able to bootstrap the undecidability of the word problem for $(\mathsf{C})\mathsf{RL}$ to prove the same for

subvarieties extended by simple equations from a class $\mathcal{U}$. This is accomplished by constructing a counter machine which is "resilient" to applications of a simple equation from $\mathcal{U}$, so-called *admissibility*, in the sense that the presence of such a simple equation does alter acceptance in the machine in a meaningful way.

4.0.1 The word problem. A *presentation* for a language $\mathcal{L}$ is a pair $\langle X, E\rangle$ where X is a set of generators and E is a set of equations over $T(X)$. A presentation $\langle X, E\rangle$ is said to be finite iff both X and E are finite. We denote the conjunction of equations in E by $\&E$. For a variety $\mathcal{V}$ of algebras in the language $\mathcal{L}$, we say $\mathcal{V}$ *has an undecidable word problem* if there exists a finite presentation $\langle X, E\rangle$ such that there is no algorithm that can decide, for inputs $s, t \in T(X)$, whether the quasi-equation

$$\&E \implies s = t \tag{4.1}$$

holds in $\mathcal{V}$. That is, membership of the set of pairs $(s, t) \in T(X)^2$ such that (4.1) holds in $\mathcal{V}$ is undecidable. Note that if $\mathcal{V}$ has undecidable word problem then its quasi-equational and universal theories are undecidable as well.

Since RL has a poset reduct, any equation $s = t$ is equivalent to the conjunction of inequations $s \leq t$ and $t \leq s$. In this way, we will consider $\leq$-rendering of the word problem

$$\&E_{\leq} \implies s \leq t, \tag{4.2}$$

where $E_{\leq}$ is a set of inequations. Consequently, for partially ordered structures, decidability of the word problem in this (inequational) rendering is equivalent to the decidability of the (equational) word problem.

By the $\{\vee, \cdot, 1\}$*-fragment of the word problem*, we mean the restriction of the word problem to inequations amongst ISR-terms in the signature $\{\vee, \cdot, 1\}$. Similarly, by the $\{\leq, \cdot, 1\}$*-fragment of the word problem*, we mean the restriction of the word problem to

inequations of monoid-terms in the signature $\{\cdot, 1\}$. Clearly, undecidability of the $\{\leq, \cdot, 1\}$-fragment implies undecidability of the $\{\vee, \cdot, 1\}$-fragment, which implies undecidability of the word problem for RL.

4.1 Algebraic Machines, Residuated frames, and the Word Problem

Let X be a finite set of variables. By $(X^*, \cdot, 1)$ we denote the free monoid generated by X, and by $\mathbf{A}_X = (A_X, \vee, \bot, \cdot, 1)$ we denote the free semiring generated by X, i.e., $(A_X, \vee)$ is a commutative semigroup, $(A_X, \cdot, 1)$ is a monoid, and $\cdot$ distributes over $\vee$. For a set $S \subseteq X^*$, define the set $S^\vee \subseteq A_X$ via

$$u \in S^\vee \iff \exists s_1, \ldots, s_n \in S,\ u =_{\mathbf{A}_X} s_1 \vee \cdots \vee s_n.$$

Note that $X^{*\vee} = A_X$.

We will call a pair $p = (x, y) \in X^* \times A_X$ an *instruction*, which we suggestively denote by $p : x \leq y$. We say an instruction $p : x \leq y$ is of *monoid-type* if furthermore $y \in X^*$. Let P be a finite set of instructions. We will call the structure $M = (X, P)$ an *algebraic machine*. The *computation relation* $\leq$ for the machine $M = (X, P)$ is defined to be the smallest $\{\cdot, \vee\}$-compatible preorder on $\mathbf{A}_X$ containing P, and will be denoted by $\leq_M$ to specify the machine M. For an instruction p, it will be useful to define the relation $\leq^p$ to be the smallest $\{\cdot, \vee\}$-compatible relation on $\mathbf{A}_X$ generated by p, i.e., the smallest relation containing p and closed under the following inference rules:

$$\frac{v \leq^p w}{xvy \leq^p xwy}\ [\cdot] \qquad \text{and} \qquad \frac{v \leq^p w}{v \vee t \leq^p w \vee t}\ [\vee],$$

for all $v, w, x, y, t \in A_X$. Consequently, $w \leq^p w'$ if and only if $p : x \leq y$ and $w = uxv \vee t$ and $w' = uyv \vee t$, for some $u, v \in X^*$ and $t \in A_X$. It is easy to verify that $\leq_M$ is equivalent to the transitive closure $\bigcup\{\leq^p : p \in P\}$. In this way, $w \leq_M w'$ iff there is a sequence of

instructions $(p_i)_{i=1}^n$ from P, and a sequence of A_X-terms $(w_i)_{i=0}^n$ such that

$$w =_{\mathbf{A}_X} w_0 \leq^{p_1} w_1 \leq^{p_2} \cdots \leq^{p_n} w_n =_{\mathbf{A}_X} w',$$

for some $n \geq 0$ called the *length of the computation*. In this way, we say $w \leq_M w'$ is witnessed by the above computation. Note that, if there is a computation witnessing $w \leq_M w'$, then there is a computation of minimal length. As a consequence of Section 1.2.4 we obtain the following proposition.[1]

Proposition 4.1.1. Let $s, t, t' \in A_X$. Then $t \vee t' \leq_M s$ if and only if there exists $s', s'' \in A_X$ such that $t \leq_M s'$ and $t' \leq_M s''$, where $s = s' \vee s''$. Furthermore, the sum of the computation lengths of $t \leq_M s'$ and $t' \leq_M s''$ is no larger than the computation length of $t \vee t' \leq_M s$.

Let $x_f \in X^*$ be a designated *final term* for M, and $\mathtt{Fin}(M) := \{x_f\}^\vee$ be the set of all finite joins of the term x_f. We will say a term $w \in A_X$ is *accepted in* M if $w \leq_M u_f$ for some $u_f \in \mathtt{Fin}(M)$, and we denote the set of all accepted terms by $\mathrm{Acc}(M) \subseteq A_X$.

Algebraic machines will typically come equipped with additional structure in which the computations are meant to operate. For instance, in the following sections on counter machines, this set is called $\mathtt{Conf}(M)$, defined depending on the type of machine M. There will typically be a designated set $Q \subseteq X$ of *states*, and a set of instructions P_{com} meant to allow states, or even other variables, to freely permute within a monoid term. If $M = (X, P \cup P_{\mathrm{com}})$, then we define $=_{\mathrm{com}}$ to be the compatible equivalence relation generated by P_{com}. We encode this form of commutativity for the instructions in P via

$$\leq^p_{\mathrm{com}} := (=_{\mathrm{com}}) \circ (\leq^p) \circ (=_{\mathrm{com}}).$$

[1] That is, $t_0 \leq_m t_1 \vee \cdots \vee t_n$ iff $\{t_1, \ldots, t_n\} \vdash_P t_0$, where $\vdash_P$ is the consequence relation generated by the set

$$\Big\{\big(\{ux_1v, \ldots, ux_kv\}, ux_0v\big) : (x_0 \leq x_1 \vee \cdots \vee x_n) \in P, u, v \in X^*\Big\}.$$

Note that if all instructions in P are monoid-type then acceptance of $w \in X^*$ reduces to $x \leq_M x_f$. We will see that the frames we construct below are defined with no mention to $\vee$, in fact the extension from $\{\leq, \cdot, 1\}$ to $\{\vee, \cdot, 1\}$ is conservative. Inspired by [14], we obtain:

Theorem 4.1.2. Let $W_M = X^*$, $W'_M = X^* \times X^*$. Then $\mathbf{W}_M = (W_M, W'_M, N_M)$ is a residuated frame, where for all $x \in W_M$ and $(u, v) \in W'_M$,

$$x \, N_M \, (u, v) \iff uxv \in \mathrm{Acc}(M)$$

Proof. We define the functions $/\!\!/, \backslash\!\!\backslash$ by $(u, v) /\!\!/ y = (u, yv)$ and $x \backslash\!\!\backslash (u, v) = (ux, v)$. Clearly, for $x, y \in W_M$ and $(u, v) \in W'_M$,

$$\begin{array}{rcccl} xy \, N_M \, (u, v) & \iff & uxyv \in \mathrm{Acc}(M) & \iff & x \, N_M \, (u, yv) \\ & \iff & y \, N_M \, (ux, v) & & \end{array} . \qquad \square$$

Lemma 4.1.3. Let $M = (X, P)$ be an algebraic machine and define the valuation $e : X \to W_M^+$ via $e(a) := \{a\}^{\triangleright\triangleleft}$. Then $\mathbf{W}_M^+, \bar{e} \models P$, where $\bar{e} : T(X) \to W_{\mathtt{B}}^+$ is the homomorphic extension of e. Furthermore, $\bar{e}(x \vee y) = \{x, y\}^{\triangleright\triangleleft}$ for any $x, y \in W_M$.

Proof. Let $\gamma := \gamma_{N_M}$ be the nucleus defined by N_M on $\wp(W_M)$, and for convenience we write $\gamma(x) := \gamma(\{x\})$ for all $x \in W_M$. Since γ is a nucleus, for each $a, b \in W_M$ we observe

$$\bar{e}(ab) = \bar{e}(a) \cdot_\gamma \bar{e}(b) = e(a) \cdot_\gamma e(b) = \gamma(\gamma(a) \cdot \gamma(b)) = \gamma(ab).$$

Hence $\bar{e}(x) = \gamma(x)$ for each $x \in W_{\mathtt{B}}$. Let $x, y \in W_{\mathtt{B}}$, then

$$\bar{e}(x \vee y) = \bar{e}(x) \cup_\gamma \bar{e}(y) = \gamma(x) \cup_\gamma \gamma(y) = \gamma(\gamma(x) \cup \gamma(y)) = \gamma(\{x, y\})$$

where the last equality is obtained using the fact that γ is expanding and idempotent.

Now, let $p \in P$ be given by $p : x \leq y$, where $y = t_1 \vee \cdots \vee t_n$ for some $x, t_1, \ldots, t_n \in W_M$. We will first show that $\{t_1, \ldots, t_n\}^{\triangleright} \subseteq \{x\}^{\triangleright}$. Suppose $(u, v) \in \{t_1, \ldots, t_n\}^{\triangleright}$, then

$$t_i \, N_M \, (u, v) \iff ut_iv \in \mathrm{Acc}(M), \; i = 1, \ldots, n.$$

By definition of $\mathrm{Acc}(M)$, this implies $uyv =_{\mathbf{A}_M} \bigvee_{i=1}^{n} ut_iv \in \mathrm{Acc}(M)$. Now $x \leq^p y$ implies $uxv \leq^p uyv$, and since $\leq_M$ is transitive it follows that $uxv \in \mathrm{Acc}(M)$. Hence $x \, N_M \, (u, v)$. So $\bar{e}(x) = \{x\}^{\triangleright\triangleleft} \subseteq \{t_1, \ldots, t_n\}^{\triangleright\triangleleft} = \bar{e}(t_1 \vee \cdots \vee t_n) = \bar{e}(y)$ and $\mathbf{W}_M^+, \bar{e} \models x \leq y$. Therefore $\mathbf{W}_M^+, \bar{e} \models P$. □

For each $w \in A_M$, we define the quasi-equation $\mathrm{acc}_M(w)$ to be

$$\& \, P \Rightarrow w \leq x_f,$$

where x_f is the final term of M.

Lemma 4.1.4. Let $\mathcal{V}$ be a subvariety of RL containing $\mathbf{W}_M^+$ for some algebraic machine $M = (X, P)$. Then for all $w \in A_M$, $w \in \mathrm{Acc}(M)$ if and only if $\mathcal{V} \models \mathrm{acc}_M(w)$.

Proof. ($\Rightarrow$) Suppose $w \in \mathrm{Acc}(M)$. By definition of $\mathrm{Acc}(M)$, $w \leq_M u_f$ for some $u_f \in \mathtt{Fin}(M)$. By definition of $\leq_M$, and there exists $n \in \mathbb{N}$, $w_0, \ldots, w_n \in A_X$, and a sequence $\{p_i\}_{i=1}^n \subseteq P$ such that

$$w = w_0 \leq^{p_1} w_1 \leq^{p_2} \cdots \leq^{p_n} w_n = u_f,$$

for some $n \geq 0$. Note that, for each $i = 1, \ldots n$, $p_i : x_i \leq y_i$ for some $x_i, y_i \in A_M$, and hence by definition on $\leq^{p_i}$

$$w_{i-1} = u_i x_i v_i \vee s_i \vee \leq^{p_i} u_i y_i v_i \vee s_i = w_i,$$

for some $u_i, v_i \in X^*$ and $s_i \in A_M$. Let $\mathbf{R} \in \mathcal{V}$ and $h : T(X) \to R$ a homomorphism. Suppose $\mathbf{R}, h \models P$. Then for each $i = 1, \ldots, n$, $h(x_i) \leq_{\mathbf{R}} h(y_i)$, and since h is a homomorphism we obtain $h(w_{i-1}) \leq_{\mathbf{R}} h(w_i)$. By transitivity of $\leq_{\mathbf{R}}$, it follows that $h(w) \leq_{\mathbf{R}} h(w_n) =_{\mathbf{R}} h(x_f)$. Since h and $\mathbf{R}$ were arbitrary, $\mathcal{V} \models \mathrm{acc}_M(w)$.

($\Leftarrow$) Let $\bar{e}$ be the map from Lemma 4.1.3. Since $\mathbf{W}_M^+, \bar{e} \models P$, and $\mathbf{W}_M^+ \in \mathcal{V}$, we have that $\mathbf{W}_M^+, \bar{e} \models w \leq x_f$. By Lemma 4.1.3, $\{w\}^{\triangleright\triangleleft} \subseteq \{x_f\}^{\triangleright\triangleleft}$, or equivalently to $\{x_f\}^{\triangleright} \subseteq \{w\}^{\triangleright}$. Since $x_f \in \mathrm{Acc}(M)$ it follows that $x_f \, N_M \, (1,1)$ by definition of N_M, so $(1,1) \in \{x_f\}^{\triangleright}$. Hence $(1,1) \in \{w\}^{\triangleright}$, i.e., $w \, N_M \, (1,1)$. Therefore by definition of N_M, $w \in \mathrm{Acc}(M)$. □

4.1.1 Complexity and the Word Problem.

As a consequence of Lemma 4.1.4, if $\mathcal{V} \subseteq \mathsf{RL}$ is a variety containing $\mathbf{W}_M^+$, for some algebraic machine $M = (X, P)$, then

$$\{\mathrm{acc}_M(u) : u \in \mathrm{Acc}(M)\} = \{\mathrm{acc}_M(u) : \mathcal{V} \models \mathrm{acc}_M(u)\} \tag{4.3}$$

Theorem 4.1.5 ([14])**.** Let M be an algebraic machine and $\mathbf{W}_M^+ \in \mathcal{V} \subseteq \mathsf{RL}$ for a variety $\mathcal{V}$. Then the computational complexity for the word problem of $\mathcal{V}$ is at least as high as the one for membership in $\mathrm{Acc}(M)$.

Proof. Suppose there is an algorithm for deciding membership in

$$Q_P = \{(s,t) \in T(X)^2 : \mathcal{V} \models \&P \implies s \leq t\}.$$

This algorithm would decide membership of the set $\{u \in T(X) : \mathcal{V} \models \mathrm{acc}_M(u)\}$. By Equation (4.3) this same algorithm would decide membership of the set $\mathrm{Acc}(M)$. $\square$

Corollary 4.1.6. If $\mathcal{V}$ is a subvariety of RL containing $\mathbf{W}_M^+$ such that membership in $\mathrm{Acc}(M)$ is undecidable, then $\mathcal{V}$ has an undecidable word problem.

Since $\{\mathrm{acc}_M(u) : \mathcal{V} \models \mathrm{acc}_M(u)\} \subseteq \{\xi : \xi \text{ is a quasi-eq. such that } \mathcal{V} \models \xi\}$, we obtain the following.

Corollary 4.1.7. Let M be an algebraic machine and $\mathcal{V} \subseteq \mathsf{RL}$ a variety. The computational complexity for the quasi-equational theory of $\mathcal{V}$ is at least as high as that of membership of $\mathrm{Acc}(M)$.

Furthermore, by Corollary 2.5.2 we obtain:

Corollary 4.1.8. Let $\mathcal{V}$ be an expansive subvariety of CRL containing $\mathbf{W}_M^+$, for some algebraic machine M. Then the computational complexity of the equational theory of $\mathcal{V}$ is as high as the one for membership of $\mathrm{Acc}(M)$.

4.1.2 Simple equations and Admissibility. Let $[\mathrm{R}] = (\mathbf{1_n}, \mathrm{R})$ be a simple equation and $M = (X, P)$ an algebraic machine. Recall that $\mathbf{W}_M^+ \models [\mathrm{R}]$ iff $\mathbf{W}_M \models (\mathrm{R})$ by Proposition 2.2.4

Define the relation $\leq^{\mathrm{R}}$ on $\mathbf{A}_X$ to be the smallest $\{\cdot, \vee\}$-compatible relation containing

$$\mathbf{1_n}^{\mathbf{A}_X}(\overline{x}) \leq \bigvee_{r \in \mathrm{R}} r^{\mathbf{A}_X}(\overline{x}),$$

for all $x_1, \ldots, x_n \in X^*$, where $\overline{x} = (x_1, \ldots, x_n)$. Equivalently, we often view $\leq^p$ as the smallest $\{\cdot, \vee\}$-compatible relation containing $\boldsymbol{\sigma}(\mathbf{1_n}) \leq \bigvee_{r \in \mathrm{R}} \boldsymbol{\sigma}(r)$, for any substitution $\boldsymbol{\sigma}$ generated by an assignment $\mathsf{Var} \to X^*$.

We view $\leq^{\mathrm{R}}$ as a sort of "ambient" instruction that can be implemented, at least trivially, to any term within a computation. In this way, it is useful to view $\leq^{\mathrm{R}}$ as a "computational glitch," and we define the relation $\leq_{\Sigma M}$ to be the computation relation generated by Σ and P for some set of simple equations Σ, i.e., $\leq_{\Sigma M}$ is the smallest $\{\cdot, \vee\}$-compatible preorder generated by $\{\leq^{\mathrm{R}}\colon [\mathrm{R}] \in \Sigma\} \cup \{\leq^{p}\colon p \in P\}$. In this way, $u \leq_{\Sigma M} u'$ iff there exists $u_0, \ldots, u_n \in A_X$ and $p_1, \ldots, p_n$ instructions such that

$$u =_{\mathbf{A}_M} u_0 \leq^{p_1} u_1 \leq^{p_2} \cdots \leq^{p_n} u_n =_{\mathbf{A}_M} u',$$

for some $n \geq 0$, where $p_i \in P \cup \Sigma$ for each $i = 1, \ldots, n$. In this way, we define $\mathrm{Acc}(\Sigma M) := \{u \in A_M : u \leq_{\Sigma M} u_f \in \mathtt{Fin}(M)\}$. Since $\leq_M \subseteq \leq_{\Sigma M}$ it easily follows that $\mathrm{Acc}(M) \subseteq \mathrm{Acc}(\Sigma M)$. Define $\mathbf{W}_{\Sigma M} = (W_M, W'_M, N_{\Sigma M})$, where $x\, N_{\Sigma M}\, u, v$ iff $uxv \in \mathrm{Acc}(\Sigma M)$

Lemma 4.1.9. Let M be an algebraic machine and Σ a set of simple equations. Then $\mathbf{W}_{\Sigma M}$ is a residuated frame and $\mathbf{W}^{+}_{\Sigma M} \models \Sigma$, i.e., $\mathbf{W}^{+}_{\Sigma M} \in \mathsf{RL} + \Sigma$.

Proof. We first observe that $\mathbf{W}_{\Sigma M}$ is a residuated frame by Theorem 4.1.2. By Proposition 2.2.4, it is enough to show that $\mathbf{W}_{\Sigma M} \models (\mathrm{R})$ for each $[\mathrm{R}] \in \Sigma$. Fix $[\mathrm{R}] = (\mathbf{1_n}, \mathrm{R}) \in \Sigma$ and suppose for some $x_1, \ldots, x_n \in W_M$ and $(u, v) \in W'_M$,

$$r^{\mathbf{A}_M}(\overline{x})\, N_{\Sigma M}\, (u, v)$$

for every $r \in \mathrm{R}$, where $\overline{x} = (x_1, \ldots, x_n)$. By definition of $N_{\Sigma M}$, it follows that $ur^{\mathbf{A}_M}(\overline{x})v \in \mathrm{Acc}(\Sigma M)$ for each $r \in \mathrm{R}$. By definition of $\leq^{\mathrm{R}}$ and since $\mathrm{Acc}(\Sigma M)$ is closed under finite joins, this implies

$$u\mathbf{1_n}^{\mathbf{A}_M}(\overline{x})v \leq^{\mathrm{R}} \bigvee_{r \in \mathrm{R}} ur^{\mathbf{A}_M}(\overline{x})v \in \mathrm{Acc}(\Sigma M).$$

Since $\mathrm{Acc}(\Sigma M)$ is closed under transitivity, it follows that $u\mathbf{1_n}^{\mathbf{A}_M}(\overline{x})v \in \mathrm{Acc}(\Sigma M)$, i.e.,

$$\mathbf{1_n}^{\mathbf{A}_M}(\overline{x})\; N_{\Sigma M}\,(u, v).$$

Hence $\mathbf{W}_{\Sigma M} \models (\mathrm{R})$. Since $[\mathrm{R}]$ was arbitrary, we obtain $\mathbf{W}^+_{\Sigma M} \models \Sigma$. □

We say Σ is *strongly admissible in* M if $\mathrm{Acc}(\Sigma M) = \mathrm{Acc}(M)$. That is, the presence of Σ-instructions does not increase the set of accepted terms. If Σ is strongly admissible in M then $\mathbf{W}^+_M = \mathbf{W}^+_{\Sigma M}$.

Corollary 4.1.10. Let M be an algebraic machine and Σ a set of simple equations. If Σ is strongly admissible in M, then $\mathbf{W}^+_M \models \Sigma$. Then for every variety $\mathcal{V}$ such that $\mathbf{W}^+_M \in \mathcal{V} \subseteq$ RL, the complexity of the word problem for $\mathcal{V}$ is at least as high as that for membership of $\mathrm{Acc}(M)$.

The following lemma will be useful for demonstrating strong admissibility, which follows as a consequence from Section 1.2.4.

Lemma 4.1.11. For a set of simple equations Σ and algebraic machine $M = (X, P)$, Σ is strongly admissible in M if and only if

$$t_1, \ldots, t_n \in \mathrm{Acc}(M) \implies t_0 \in \mathrm{Acc}(M),$$

for all $t_0, \ldots, t_n \in X^*$ such that $t_0 \leq^{\mathrm{R}} t_1 \vee \cdots \vee t_n$, for some $[\mathrm{R}] \in \Sigma$.

4.1.3 Canonically admissible. For the machines we use there will typically be some proper subset of X^* of *canonical monoid terms* on which operations of the machine remain *stable*. Let $C \subseteq X^*$. We say a set $C \subseteq X^*$ is *stable in* M if $x_f \in C$ and for all $p \in P$ and $u, v \in A_X$, if $u \leq^p$ then $u \in C^\vee$ iff $v \in C^\vee$.

For a given stable set C, we will often use typewriter font to denote elements, e.g., $\mathtt{C} \in C$ and $\mathtt{u} \in C^\vee$.

Proposition 4.1.12. Let $M = (X, P)$ be an algebraic machine with set C stable in M.

1. Suppose $u \leq_M v$. Then $u \in C^\vee$ iff $v \in C^\vee$.
2. $\mathrm{Acc}(M) \subseteq C^\vee$.

Proof. Since C is stable in M, this follows by induction on computation length. □

We now introduce a weaker notion of admissibility, called *canonically admissible*. Suppose C is stable in M, and let $\mathtt{Can}(M) = C^\vee$. By Proposition 4.1.12, $\mathrm{Acc}(M) \subseteq \mathtt{Can}(M)$. However, for a given set of simple equations Σ, it is possible that $\mathrm{Acc}(\Sigma M) \nsubseteq \mathtt{Can}(M)$ (e.g., Section 5.1.1). In which case there must exist some $u, v \in A_M$ and $[\mathrm{R}] \in \Sigma$ such that $u \leq^{\mathrm{R}} v$ but $u \notin \mathtt{Can}(M)$ and $v \in \mathtt{Can}(M)$, and so Σ is not strongly admissible in M.

We say Σ is *canonically admissible in* M if for every $u \in \mathtt{Can}(M)$,

$$u \in \mathrm{Acc}(\Sigma M) \iff u \in \mathrm{Acc}(M).$$

We will often refer to canonical admissibility simply by *admissible*. This means that the presence of Σ does not increase acceptance in a meaningful way, modulo the canonical terms. It is clear that strong admissibility implies admissibility.

Theorem 4.1.13. Let M be an algebraic machine such that $\mathrm{Acc}(M) \subseteq \mathtt{Can}(M)$. Let Σ be a set of simple equations such that $\mathbf{W}^+_{\Sigma M} \in \mathcal{V}$ for some variety $\mathcal{V} \subseteq \mathsf{RL}$. If Σ is admissible in M then the complexity of the word problem for $\mathcal{V}$ is at least as high as that for $\mathrm{Acc}(M)$.

4.1.4 Hardware-admissibility. The algebraic machines $M = (X, P)$ we consider will have designated sets $H_1, \ldots, H_k \subseteq X$ of *hardware*.[2] Let $H \subseteq X$ be a set of *hardware*

[2]The hardware that we consider later will always consist of a set H_0 of *states*, and in some cases a sequence $i = 1, \ldots, k+1$ of stopper variables, where $H_i = \{\mathtt{S}_i\}$.

with elements called *components*. We say M is ***H*-stable** if the final term $x_f \in X^*$ for M contains precisely one component variable from H, and for each instruction $p \in P$, where $p : t_0 \leq t_1 \vee \cdots \vee t_n$ for some $t_0, t_1, \ldots, t_n \in X^*$, $t_i \in X^*$ contains precisely one component variable for each $i = 0, \ldots, n$.[3] Let $H_X^* \subseteq X^*$ be the set of all monoid terms that contain precisely one component variable from H. That is, $H_X^* = (X \setminus H)^* \cdot H \cdot (X \setminus H)^*$. Clearly, if M is H-stable then the set H_X^* is stable for M,

Let Σ be set of simple equations and $M = (X, P)$ be H-stable with hardware $H \subseteq X$. We say Σ is ***H*-admissible** in M if $\mathrm{Acc}(\Sigma M) \subseteq H_X^{*\vee}$.

Recall that Σ is mingly iff Σ contains a simple equation $[\mathrm{R}]$ that is either integral or $[\boldsymbol{\sigma}\mathrm{R}] : x^k \leq x$ for some $k > 1$ and substitution $\boldsymbol{\sigma}$.[4]

Lemma 4.1.14. Let $M = (X, P)$ be an H-stable algebraic machine with some set of hardware H, and Σ a set of simple equations. The following are equivalent:

1. Σ is H-admissible in M.
2. For every $w, w' \in A_X$, $w \leq_{\Sigma m} w'$ implies $w \in H_X^{*\vee} \iff w' \in H_X^{*\vee}$.
3. Σ is not mingly.

Proof. Let x_f be the final term for M.

(1 $\Rightarrow$ 2) Assume $\mathrm{Acc}(\Sigma M) \subseteq H_X^{*\vee}$. Since H_X^* is stable in M, $\mathtt{Fin}(M) \subseteq H_X^{*\vee}$. By the same argument as Proposition 4.1.12, it follows that $w \leq_{\Sigma M} w'$ implies $w \in H_X^{*\vee} \iff w' \in H_X^{*\vee}$.

(2 $\Rightarrow$ 3) Proceeding by contraposition, suppose $[\mathrm{R}] = (\mathbf{1_n}, \mathrm{R}) \in \Sigma$ is mingly. Then there is a 1-variable substitution $\boldsymbol{\sigma}$ such that $\boldsymbol{\sigma}(\mathbf{1_n}) = x^k$ and $\boldsymbol{\sigma}[\mathrm{R}] = \{x^\delta\}$ where $\delta \in$

[3] That is, $t \in X^*$ contains precisely one component variable iff $\mathrm{supp}(t) \cap H = \{q\}$ and $\#(t, q) = 1$.

[4] See Theorem 2.4.1.

$\{0,1\}$ and $k > \delta$. This implies that

$$x_f^{k+\delta'} \leq^{\mathrm{R}} \bigvee_{r\in\mathrm{R}} x_f^{\delta+\delta'} = \bigvee_{r\in\mathrm{R}} x_f \in \mathtt{Fin}(M),$$

where $\{\delta,\delta'\} = \{0,1\}$. Since M is H-stable, $\mathtt{Fin}(M) \subseteq H_X^{*\vee}$. Since $k+\delta' > 1$, $x_f^{k+\delta'} \notin H_X^*$. Hence $w \leq_{\Sigma M} w'$ such that $w \notin H_X^{*\vee}$ but $w' \in H_X^{*\vee}$.

$(3 \Rightarrow 1)$ Proceeding by contraposition, suppose there $\mathrm{Acc}(\Sigma M) \setminus H_X^{*\vee} \neq \emptyset$. We may assume $t \in \mathrm{Acc}(\Sigma M) \setminus H_X^{*\vee}$ has the minimal computation length $N \geq 0$ witnessing this fact. I.e., if $t \leq_{\Sigma M} \mathtt{u}_f \in \mathtt{Fin}(M)$ has a witness of length N. By Proposition 4.1.1, the minimality of N implies $t \in X^* \setminus H_X^*$. Since $\mathtt{Fin}(M) \subseteq H_X^{*\vee}$ by assumption, it follows that $N > 1$. So $t \leq^p w \leq_{\Sigma M} \mathtt{u}_f$ for some instruction $p \in P \cup \{\mathrm{R} : [\mathrm{R}] \in \Sigma\}$ and $w \in X^{*\vee}$. Since N is minimal and $w \in \mathrm{Acc}(\Sigma M)$, it follows that $w \in H_X^{*\vee}$ and hence $p \in \{\mathrm{R} : [\mathrm{R}] \in \Sigma\}$ since H_X^* is stable. Hence there is $[\mathrm{R}] = (\mathbf{1_n}, \mathrm{R}) \in \Sigma$ such that

$$t = u\mathbf{1_n}(\overline{x})v \leq^{\mathrm{R}} \bigvee_{r\in\mathrm{R}} ur(\overline{x})v = w$$

for some $u, v, x_1, \ldots, x_n \in X^*$ where $\overline{x} = (x_1, .., x_n)$. Define $I := \{i \leq n : x_i \in X^* \setminus (X \setminus H)*\}$ to be the set of all $i \leq n$ such that x_i contains at least one component variable. We have two cases:

Case 1: If $uv \in H_X^*$, then $r(\overline{x}) \in (X \setminus H)^*$ for all $r \in \mathrm{R}$. Hence for all $i \in I$, $\#(r, y_i) = 0$ for each $r \in \mathrm{R}$, where $\mathrm{supp}(\mathbf{1_n}) = \{y_1, \ldots, y_n\}$. Without loss of generality, suppose $1 \in I$. Consider the 1-variable substitution $\boldsymbol{\sigma}$ that maps $y_1 \mapsto y$ and $y_j \mapsto 1$ for all $j > 1$. Then $[\boldsymbol{\sigma}\mathrm{R}] : x \leq 1$. Hence $[\mathrm{R}]$ is mingly.

Case 2: If $uv \notin H_X^*$, then $uv \in (X \setminus H)*$, otherwise $w \notin H_X^{*\vee}$. Since Σ is a set of simple equations, it is non-degenerate and so every variable that appears in w must appear in t. Since $w \in H_X^{*\vee}$, it must be that $\overline{x}$ contains a component variable. Hence for each

$r \in \mathrm{R}$, there exists a unique $i_r \in I$ such that $\#(r, y_{i_r}) = 1$ and for each $i \in I \setminus \{i_r\}$, $\#(r(, y_i) = 0$, otherwise $ur(\overline{x})v \notin H_X^*$. We claim that $|I| > 1$. Suppose otherwise, i.e. $I = \{i\}$, and thus $\#(r, y_i) = 1$ for all $r \in \mathrm{R}$. But this implies $x_i \in H_X^*$ and $x_j \in (X \setminus H)^*$ for all $j \neq i$ since $ur(\overline{x})v \in H_X^*$ for all $r \in \mathrm{R}$. But this implies

$$t = u\mathbf{1_n}(\overline{x})v = ux_1 \cdots x_n v \in H_X^*,$$

since $u, v, x_j \in (X \setminus H_X^*)^*$, a contradiction.

Hence $k := |I| \geq 2$. Consider the substitution $\boldsymbol{\sigma}$ that maps $y_i \mapsto y$ for all $i \in I$, otherwise and $y_j \mapsto 1$ for $j \notin I$. Then $\boldsymbol{\sigma}(r) = \boldsymbol{\sigma}(y_{i_r}) = y$ for all $r \in \mathrm{R}$, and $\boldsymbol{\sigma}(\mathbf{1_n}) = y^k$. Hence $[\boldsymbol{\sigma}\mathrm{R}] : y^k \leq y$. Therefore $[\mathrm{R}]$ is mingly. □

4.2 Counter machines in RL and the $\{\leq, \cdot, 1\}$-fragment

For proving lower-bounds for the complexity of decision problems, we will use a type of abstract machine known as a *Counter Machine* (CM). We will first present CM's in a semi-informal language typical for such automata, and then present two representations of counter machines as algebraic machines, as defined in the previous section. A more detailed exposition of counter machines as multi-tape Turing machines can be found in [13].

A CM models a computer having a finite number of *registers* $\mathtt{r}_1, \ldots, \mathtt{r}_k$ each of which can contain an arbitrary non-negative integer (a machine with k registers will be called a k-CM), a finite number of *states* with a designated final state q_f, and a finite set of *instructions* that indicate whether, given a certain state of the machine, to alter the contents of the registers, in a specific way, and update its state. The basic instructions are of the machine are called *increment*, *decrement*, and *zero-test*, which consist of consist of adding 1 to a register, subtracting 1 from a non-empty register, or verifying a register is empty, respectively.

We will represent a k-CM $\mathtt{M}$ by the triple $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P})$, where $\mathtt{R}_k = \{\mathtt{r}_1, \ldots, \mathtt{r}_k\}$ is a set of register-names, $\mathtt{Q}$ is a finite set of states with designated final state q_f, and $\mathtt{P}$ is a finite set of basic instructions, with no instruction of the form $q_f \cdots$ (i.e., q_f is only an outgoing transitional state). Using a fresh set of variables $\mathtt{Stp}_k := \{\mathtt{S}_1, \ldots, \mathtt{S}_k, \mathtt{S}_{k+1}\}$, define $(W_{\mathtt{M}}, \cdot, 1)$ to be the free monoid generated by $\mathtt{R}_k \cup \mathtt{Stp}_k \cup \mathtt{Q}$.

A *configuration* of a k-CM is the description of the machines current state and register contents, and will be canonically associated with a monoid word

$$q\mathtt{S}_1\mathtt{r}_1^{n_1}\mathtt{S}_2\mathtt{r}_2^{n_2}\mathtt{S}_3 \cdots \mathtt{S}_k\mathtt{r}_k^{n_k}\mathtt{S}_{k+1}$$

The data-type of a configuration is essentially that of a tuple $\langle q; n_1, \ldots, n_k \rangle$, where q indicates the current state of the machine, and $n_1, \ldots, n_k \in \mathbb{N}$ are the current contents of registers $\mathtt{r}_1, \ldots, \mathtt{r}_k$, respectively.

The instructions of a machine will be written as $q\mathtt{S}_i \leq q'\mathtt{S}_i\mathtt{r}_i$, $q\mathtt{S}_i\mathtt{r}_i \leq q'\mathtt{S}_i$, and $q\mathtt{S}_i\mathtt{S}_{i+1} \leq q'\mathtt{S}_i\mathtt{S}_{i+1}$, for some states q, q' and $i \in \{1, \ldots, k\}$, denoting instances of increment $\mathtt{r}_i$, decrement $\mathtt{r}_i$, and zero-test $\mathtt{r}_i$ instructions. In this way, the basic instructions of a CM are understood as follows: when the machine is in state q and there is an increment instruction $q\mathtt{S}_i \leq q'\mathtt{S}_i\mathtt{r}_i$, then the machine may increment the register $\mathtt{r}_i$ by 1 and transition to the state q'; when the machine is in state q and there is a decrement instruction $q\mathtt{S}_i\mathtt{r}_i \leq q'\mathtt{S}_i$, then if the value of the register $\mathtt{r}_i$ is nonzero the machine may decrement the register $\mathtt{r}_i$ by 1 and transition to the state q'; when the machine is in state q and there is a zero-test instruction $q\mathtt{S}_i\mathtt{S}_{i+1} \leq q'\mathtt{S}_i\mathtt{S}_{i+1}$, then if the register $\mathtt{r}_i$ is empty, i.e., has value 0, the machine may transition to the state q' with the registers unaltered.

For each $p \in \mathtt{P}$, define $\leq^p$ to be the compatible relation on $W_{\mathtt{M}}$ defined as in Section 4.1. i.e., as the relation on $W_{\mathtt{M}}$ containing p and closed under the inference rule

$$\frac{u \leq^p v}{xuy \leq^p xvy}\ [\cdot], \tag{4.4}$$

for all $u, v, x, y \in W_{\mathtt{M}}$.

To implement instructions as intended, we must allow the state variables to freely permute within terms from $W_{\mathtt{B}}$. We define the set

$$\mathtt{P}_{\mathrm{com}} := \{qx \leq xq : q \in \mathtt{Q}, x \in \mathtt{R}_k \cup \mathtt{Stp}_k\} \cup \{xq \leq qx : q \in \mathtt{Q}, x \in \mathtt{R}_k \cup \mathtt{Stp}_k\}, \tag{4.5}$$

and the equivalence relation $=_{\mathrm{com}}$ defined as in Section 4.1. We will abuse notation and write $\leq^p$ instead of $\leq^p_{\mathrm{com}}$ if the context is clear. We note that, since all instructions are of monoid type, the closure under the inference rule $[\vee]$ is not needed in this context. However, the closure of $[\vee]$ is conservative, and will be useful for the following chapter.

For a given k-CM $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P})$, the structure $(X_{\mathtt{M}}, P)$ is an algebraic machine, where $X_{\mathtt{M}} = \mathtt{R}_k \cup \mathtt{Stp}_k \cup \mathtt{Q}$ and $P_{\mathtt{M}} = \mathtt{P} \cup \mathtt{P}_{\mathrm{com}}$. We will abuse notation by denoting $(X_{\mathtt{M}}, P_{\mathtt{M}})$ by $\mathtt{M}$. In this way, the computation relation $\leq_{\mathtt{M}}$ is as defined in Section 4.1. As before, it will be useful to view $\leq_{\mathtt{M}}$ as the transitive closure of $(=_{\mathrm{com}}) \cup \{\leq^p : p \in \mathtt{P}\}$.

Define the set of *register boxes*

$$\mathtt{Box}_k := \{\mathtt{S}_1\mathtt{r}_1^{n_1}\mathtt{S}_2\mathtt{r}_2^{n_2}\mathtt{S}_3 \cdots \mathtt{S}_k\mathtt{r}_k^{n_k}\mathtt{S}_{k+1} \in W_{\mathtt{M}} : n_1, \ldots, n_k \in \mathbb{N}\}$$

and the set of *configurations* by

$$\mathtt{Conf}(\mathtt{M}) := \{uqv \in W_{\mathtt{M}} : q \in \mathtt{Q}, uv \in \mathtt{Box}_k\},$$

and by $\mathtt{C}_f := q_f\mathtt{S}_1 \cdots \mathtt{S}_k\mathtt{S}_{k+1}$ we denote the *final configuration*, where $q_f \in \mathtt{Q}$ is the final state of $\mathtt{M}$. Now, we see that for given configurations $\mathtt{C}, \mathtt{C}' \in \mathtt{Conf}(\mathtt{M})$,

$$\mathtt{C} =_{\mathrm{com}} \mathtt{C}' \iff \mathtt{C} = uqv \;\&\; \mathtt{C}' = u'qv' \;\&\; uv = u'v' \in \mathtt{Box}_k \ , \tag{4.6}$$

Abusing notation, we write $\mathtt{C} = \langle q; n_1, \ldots, n_k\rangle$ iff $\mathtt{C} = uqv \in \mathtt{Conf}(\mathtt{M})$ and $uv = \mathtt{S}_1 \prod_{i=1}^{k} \mathtt{r}_i^{n_i}\mathtt{S}_{i+1}$. Note that $\mathtt{C} =_{\mathtt{Q}} \mathtt{C}_f$ iff $\mathtt{C} = \langle q_f; 0, \ldots, 0\rangle$.

In this light, we view implementations of instructions from $\mathtt{P}$ as follows:

$p : q\mathtt{S}_i \leq q'\mathtt{S}_i\mathtt{r}_i$	$\langle q; n_1, \ldots, n_i, \ldots, n_k\rangle$	$\leq^p_{\mathrm{com}}$	$\langle q'; n_1, \ldots, n_i + 1, \ldots n_k\rangle$
$p : q\mathtt{S}_i\mathtt{r}_i \leq q'\mathtt{S}_i$	$\langle q; n_1, \ldots, n_i + 1, \ldots, n_k\rangle$	$\leq^p_{\mathrm{com}}$	$\langle q'; n_1, \ldots, n_i, \ldots n_k\rangle$
$p : q\mathtt{S}_i\mathtt{S}_{i+1} \leq q'\mathtt{S}_i\mathtt{S}_{i+1}$	$\langle q; n_1, \ldots, n_{i-1}, 0, \ldots, n_k\rangle$	$\leq^p_{\mathrm{com}}$	$\langle q'; n_1, \ldots, n_{i-1}, 0, \ldots, n_k\rangle$

It easily follows that $\mathtt{C} \leq_{\mathtt{M}} \mathtt{C}'$ if and only if there exists $n \geq 0$, $\mathtt{C}_0, \ldots, \mathtt{C}_n \in \mathtt{Conf}(\mathtt{M})$, and a sequence of instructions $(p_i)_{i=1}^n$ from $\mathtt{P}$ such that

$$\mathtt{C} = \mathtt{C}_0 \leq^{p_1}_{\mathrm{com}} \cdots \leq^{p_n}_{\mathrm{com}} \mathtt{C}_n = \mathtt{C}'.$$

The following will serve as our undecidable problem:

Proposition 4.2.1 ([20, 16])**.** There exists a 2-CM $\tilde{\mathtt{M}}$ for which membership of $\mathrm{Acc}(\tilde{\mathtt{M}})$ is undecidable.

Example 4.2.1. Consider the 1-CM $\mathtt{M}_{\mathrm{even}} = (\mathtt{R}_1, \mathtt{Q}_{\mathrm{even}}, \mathtt{P}_{\mathrm{even}})$, where $\mathtt{Q}_{\mathrm{even}} = \{q_0, q_1, q_f\}$ and $\mathtt{P}_{\mathrm{even}} = \{p_0, p_1, p_f\}$ are given by

$$\begin{aligned} p_0 &: q_0\mathtt{S}_1\mathtt{r}_1 \leq q_1\mathtt{S}_1 \\ p_1 &: q_1\mathtt{S}_1\mathtt{r}_1 \leq q_0\mathtt{S}_1 \\ p_f &: q_0\mathtt{S}_1\mathtt{S}_2 \leq q_f\mathtt{S}_1\mathtt{S}_2. \end{aligned}$$

For an example,

$$q_0\mathtt{S}_1\mathtt{r}_1^2\mathtt{S}_2 \leq^{p_0} q_1\mathtt{S}_1\mathtt{r}_1\mathtt{S}_2 \leq^{p_1} q_0\mathtt{S}_1\mathtt{S}_2 \leq^{p_f} q_f\mathtt{S}_1\mathtt{S}_2$$

is a computation showing that $\langle q_0; 2\rangle$ is accepted in $\mathtt{M}_{\text{even}}$. On the other hand, the only computation possible starting from the configuration $\langle q_0; 1\rangle$ is given by $\langle q_0; 1\rangle \sqsubseteq^{p_0} \langle q_1; 0\rangle$, and so $\langle q_0; 1\rangle$ is not accepted. In general, it is easy to see that $\langle q_0; n\rangle$ is accepted in $\mathtt{M}_{\text{even}}$ if and only if n is even.

4.2.1 Counter machines and residuated frames. Consider an instruction $p : u \leq v$ in $\mathtt{P} \cup \mathtt{P}_{\text{com}}$ from definitions in Equations (4.4) and (4.5). We observe that $u, v \in W_{\mathtt{M}}$, i.e., the instruction p is of monoid type, and each contain precisely one state-variable. Furthermore, the terms u, v contain precisely the same stopper variables with the same multiplicity, where no stopper variable has multiplicity greater than 1, and no stopper variables in u are permuted in v. Since $\mathtt{C}_f \in \mathtt{Conf}(\mathtt{M})$, by the above the following is immediate:

Lemma 4.2.2. The set $\mathtt{Conf}(\mathtt{M})$ is stable in $\mathtt{M}$. In particular, $\mathtt{M}$ is H-stable for all $H \in \{\mathtt{Q}, \{\mathtt{S}_1\}, \ldots \{\mathtt{S}_{k+1}\}\}$.

By Corollary 4.1.6 and the fact that all instructions in $\mathtt{P} \cup \mathtt{P}_{\text{com}}$ are of type $\{\cdot, 1\}$, we obtain:

Theorem 4.2.3. Let $\mathtt{M}$ be a k-CM and $\mathbf{W}_M^+ \in \mathcal{V} \subseteq \mathsf{RL}$ for a variety $\mathcal{V}$. Then the computational complexity for even the $\{\leq, \cdot, 1\}$-fragment word problem of $\mathcal{V}$ is at least as high as the one for membership in $\mathrm{Acc}(\mathtt{M})$.

By Proposition 4.2.1, since membership of $\mathrm{Acc}(\tilde{\mathtt{M}})$ is undecidable, we obtain:

Corollary 4.2.4. Any variety $\mathcal{V} \subseteq \mathsf{RL}$ for which $\mathbf{W}_{\tilde{\mathtt{M}}}^+ \in \mathcal{V}$, the word problem, particularly for the $\{\leq, \cdot, 1\}$-fragment, of $\mathcal{V}$ is undecidable.

4.2.2 Observations on admissibility. We first observe that $\mathbf{W}^+_{\mathtt{M}} \notin \mathsf{CRL}$ for any CM $\mathtt{M}$. Let $u = \mathtt{S}_1$, $v = \mathtt{S}_2$, and $w = \prod_{i=2}^{k} \mathtt{S}_{i+1}$. Clearly, the final configuration $\mathtt{C}_f = q_f uvw \in \mathrm{Acc}(\mathtt{M})$ but $q_f vuw \notin \mathrm{Acc}(\mathtt{M})$ since it is not a configuration. In other words, $uv\, N_{\mathtt{M}}(q_f, w)$ but $vu \not\!N_{\mathtt{M}}(q_f, w)$, and hence by Proposition 2.2.4 $\mathbf{W}^+_{\mathtt{M}} \not\models xy \leq yx$ for any k-CM $\mathtt{M}$.

In fact, by Mayr and Meyer [19], the $\{\leq, \cdot, 1\}$-fragment of CRL has a decidable quasi-equational theory, and therefore this particular algebraic rendering of counter-machines is insufficient to capture undecidability for commutative varieties. In the next section, we will present an algebraic rendering in which commutativity will be admissible, at the cost of adding $\vee$ to the signature.

On the other hand, $\mathbf{W}^+_{\mathtt{M}}$ satisfies the permutation of squares $x^2y^2 = y^2x^2$. In essence, this due to the fact,

$$\begin{aligned}
x^2y^2\, N_{\mathtt{M}}(u,v) &\iff ux^2y^2v \in \mathrm{Acc}(\mathtt{M}) && [\text{by def. of } N_{\mathtt{M}}]\\
&\implies x^2y^2 \text{ contains no variables in } \mathtt{Q} \cup \mathtt{Stp}_k && [\mathrm{Acc}(\mathtt{M}) \subseteq \mathtt{Conf}(\mathtt{M})]\\
&\implies x^2y^2 = \mathtt{r}_i^n \text{ for some } i \leq k \text{ and } n \in \mathbb{N} && [\text{by def. of } \mathtt{Conf}(\mathtt{M})]\ .\\
&\implies x^2y^2 = y^2x^2\\
&\implies y^2x^2\, N_{\mathtt{M}}(u,v),
\end{aligned}$$

Let $[\mathrm{R}] = (\mathbf{1_4}, \mathrm{R})$ be the linearization of $x^2y^2 \leq y^2x^2$,

$$[\mathrm{R}] : x_1x_2y_1y_2 \leq \bigvee\{y_iy_jx_lx_k : 1 \leq i,j,k,l \leq 2\}.$$

Since each $y^2x^2 \in \mathrm{R}$ for each $y \in \{y_1, y_2\}$ and $x \in \{x_1, x_2\}$, the above argumentation establishes that $[\mathrm{R}]$ is strongly admissible in $\mathtt{M}$, so $\mathbf{W}_{\mathtt{M}} \models (\mathrm{R})$, and hence $\mathbf{W}^+_{\mathtt{M}} \models [\mathrm{R}]$. More generally, this same argument shows $\mathbf{W}^+_{\mathtt{M}} \models x^ny^m \leq y^mx^n$ for any integers $m, n \geq 2$.[5]

[5]Note that $x^ny^m \leq y^mx^n$ is RL-equivalent to $x^ny^m = y^mx^n$.

Therefore, by Corollary 4.1.10, the word problem for $\mathsf{RL} + (x^n y^m = y^m x^n)$, in particular its $\{\leq, \cdot, 1\}$-fragment, is undecidable for any $m, n \geq 2$.

The above argument also follows as a consequence from the following technical lemma, which will be needed in Section 5.2.2. For a set Σ of simple equations, we say Σ *entails commutativity* if $\mathsf{ISR} + \Sigma$ is commutative, or equivalently $yx \vdash_\Sigma xy$ by Theorem 2.3.4.

Lemma 4.2.5. Let Σ be a non-mingly set of simple equations and $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P})$ a counter machine. If Σ does not entail commutativity then for a monoid term t, if $t \in \mathrm{Acc}(\Sigma\mathtt{M})$ then $t = uqv$ where $q \in \mathtt{Q}$, and

$$uv = \mathtt{S}_1 x_1 \mathtt{S}_2 x_2 \cdots \mathtt{S}_k x_k \mathtt{S}_{k+1}, \text{ with } x_1, \ldots, x_k \subseteq \mathtt{R}_k^*. \tag{4.7}$$

Proof. Suppose $t \in \mathrm{Acc}(\Sigma\mathtt{M})$. Let $H \in \{\mathtt{Q}_K, \{\mathtt{S}_1\}, \ldots, \{\mathtt{S}_k\}, \{\mathtt{S}_{k+1}\}\}$. By Lemma 4.2.2, $\mathtt{M}$ is H-stable. By Lemma 4.1.14, Σ is H-admissible since Σ is not mingly. Therefore t contains precisely one state variable from H. Since this holds for all sets H, $t = uqv$ for some $q \in \mathtt{Q}$ and

$$uv = \mathtt{S}_{n_1} x_1 \mathtt{S}_{n_2} x_2 \cdots \mathtt{S}_{n_k} x_k \mathtt{S}_{n_{k+1}},$$

where $x_1, \ldots, x_k \subseteq \mathtt{R}_k^*$ and $\{1, \ldots, k+1\} = \{n_1, \ldots, n_{k+1}\}$. We need only show that if Σ does not entail commutativity then $n_i = i$ for each $i = 1, \ldots, k+1$. We induct on the minimal computation length $N \geq 0$ witnessing $t \leq_{\Sigma\mathtt{M}} u_f \in \mathtt{Fin}(\Sigma\mathtt{M})$. Clearly, if $N = 0$, then $t = \mathtt{C}_f = q_f \mathtt{S}_1 \cdots \mathtt{S}_k \mathtt{S}_{k+1}$ and we are done. So suppose the claim holds for all computations of length $M < N$. Then $t \leq^p \bigvee_{j=1}^m t_i \in \mathrm{Acc}(\Sigma\mathtt{M})$ for some instruction $p \in \mathtt{P} \cup \Sigma$ and monoid terms $t_1, \ldots, t_m$. Hence by the induction hypothesis, the term t_j has the form of Equation (4.7), for all $j = 1, \ldots, m$. Now, if $p \in \mathtt{P}$ then $m = 1$. Since no instruction in $\mathtt{P}$ permutes stopper variables, it follows that $n_i = i$ for each $i = 1, \ldots, k+1$.

So we may assume $p = [\mathrm{R}] = (\mathbf{1_n}, \mathrm{R}) \in \Sigma$. Hence $\{t_1, \ldots, t_m\} = \{t_r : r \in \mathrm{R}\}$, and

$$t = w\boldsymbol{\sigma}(\mathbf{1_n})w' \leq^{\mathrm{R}} \bigvee_{r\in\mathrm{R}} w\boldsymbol{\sigma}(r)w',$$

for some substitution $\boldsymbol{\sigma}$ and monoid terms w, w'.

Suppose the contrary, i.e., that $n_i \neq i$ for some $i \leq k+1$. We will show $[\mathrm{R}]$ entails commutativity. Without loss of generality, we may assume t is of the form $t = a\mathtt{S}_2 b\mathtt{S}_1 c$. Since each t_i is of the form of 4.7, it follows that $\mathtt{S}_2 b\mathtt{S}_1$ must be a subword of $\boldsymbol{\sigma}(\mathbf{1_n})$. Now, if either $\mathtt{S}_1$ or $\mathtt{S}_2$ are subwords of ww', then this implies $\mathrm{supp}(\mathbf{1_n}) \setminus \mathrm{supp}(\mathrm{R}) \neq \emptyset$, making $[\mathrm{R}]$ integral. Since $[\mathrm{R}]$ is non-mingly, it follows that neither $\mathtt{S}_1$ not $\mathtt{S}_2$ are subwords of ww'. Hence there exists variables $x, y \in \mathrm{supp}(\mathbf{1_n})$ such that $\mathtt{S}_2$ is a subword of $\boldsymbol{\sigma}(x)$ and $\mathtt{S}_1$ is a subword of $\boldsymbol{\sigma}(y)$, and $x, y \in \bigcap_{r\in\mathrm{R}} \mathrm{supp}(r)$. Note x appears to the left of y in $\mathbf{1_n}$. For each $r \in \mathrm{R}$, since t_r has form 4.7, it follows that $\#(r, x) = \#(r, y) = 1$ and y appears to the left of x in r. Consider the substitution $\boldsymbol{\tau}$ that maps $x \mapsto x$, $y \mapsto y$, and for all variables $z \notin \{x, y\}$, $z \mapsto 1$. Then

$$xy = \boldsymbol{\tau}(\mathbf{1_n}) \leq^{[\mathrm{R}]} \bigvee_{r\in\mathrm{R}} \boldsymbol{\tau}(r) = \bigvee_{r\in\mathrm{R}} yx = yx.$$

Hence $[\mathrm{R}]$ entails commutativity. □

We will revisit admissibility of simple equations for such structures in a Section 5.2.2.

4.3 And-branching counter machines in (C)RL and the $\{\vee, \cdot, 1\}$-fragment

We now define a class of machines known as an *And-branching k-Counter Machine* (k-ACM), as introduced in [17] to prove the undecidability of linear logic. A k-ACM is essentially the same as a k-CM with the exception that there are no zero-test instructions, but rather "branching" instructions that are typically called *forking*. In this way, a k-ACM is a type of parallel-computing counter machine where instructions replace a configuration by

a possible set of configurations, and the machine involved computes one finite set of configurations from another. For our purposes, a k-ACM is a tuple $\mathtt{B} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P})$ representing a type of parallel-computing counter machine, where

- $\mathtt{R}_k := \{\mathtt{r}_1, \ldots, \mathtt{r}_k\}$ is a set of k *registers*, each able to store a non-negative integer (representing the number of tokens in that register),
- $\mathtt{Q}$ is a finite set of *states* with a designated *final state* q_f,
- and $\mathtt{P}$ is a finite set of *instructions* (to be formalized below) that indicate whether to, given a certain state of the machine, *increment* a register or *decrement* a nonzero register, as well as a "branching" instruction known as *forking*, with no instruction applicable to the state q_f.

The most important feature of ACMs is their ability to capture some effect of the zero-tests in the presence of commutativity, as we will see in Section 4.3.2. In the case for CMs, if $\mathtt{P}_{\mathrm{com}}$ contained all variable pairs, e.g., $\leftarrow : \mathtt{S}_i\mathtt{r}_i \leq \mathtt{r}_i\mathtt{S}_i$ and $\rightarrow : \mathtt{r}_i\mathtt{S}_i \leq \mathtt{S}_i\mathtt{r}_i$, then improper implementations of a zero-test $p : \mathtt{S}_i q \mathtt{S}_{i+1} \leq \mathtt{S}_i q' \mathtt{S}_{i+1}$ as follows,

$$q\mathtt{S}_i\mathtt{r}_i\mathtt{S}_{i+1} \leq^{\leftarrow} q\mathtt{r}_i\mathtt{S}_i\mathtt{S}_{i+1} =_{\mathrm{com}} \mathtt{r}_i\mathtt{S}_i q\mathtt{S}_{i+1} \leq^{p} \mathtt{r}_i\mathtt{S}_i q\mathtt{S}_{i+1} =_{\mathrm{com}} q'\mathtt{r}_i\mathtt{S}_i\mathtt{S}_{i+1} \leq^{\rightarrow} q'\mathtt{S}_i\mathtt{r}_i\mathtt{S}_{i+1}.$$

Since we will allow $\mathtt{P}_{\mathrm{com}}$ to contain such instructions, the stopper variables can all be pushed to the back, and are therefore irrelevant. Therefore, a *configuration* $\mathtt{C}$ of a k-ACM coincides with that of a k-CM, i.e., it is a tuple consisting of a single state and, for each register, a nonnegative integer indicating the contents of that register, but with the stopper variables removed. We represent a configuration $\mathtt{C}$ as a term in the free monoid generated by $\mathtt{Q} \cup \mathtt{R}_k$, and canonically arranged as

$$q\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\cdots\mathtt{r}_k^{n_k},$$

We imagine a configuration being a box labeled by a state and containing tokens labelled by elements from the set $\mathtt{R}_k$. where $q \in \mathtt{Q}$ is the *state* of the configuration and n_i is the number stored in the register $\mathtt{r}_i$, for each $i = 1, .., k$, and if $n_i = 0$, we say the register $\mathtt{r}_i$ is *empty*. Since $\mathtt{C}$ contains precisely one state, we canonically identify configurations with the set $\mathtt{Q} \cdot \mathtt{R}_k^*$.

The instructions of a k-ACM replace a single configuration by a new configuration (via increment and decrement), or by two configurations (via forking). The increment and decrement instructions will be given by $q \leq q'\mathtt{r}_i$ and $q\mathtt{r}_i \leq q'$, respectively, and are understood as per usual. A *forking* instruction will be of the form $q \leq q' \vee q''$, and can be understood as "if a box is labeled by state q, duplicate the box and its contents, resulting in two boxes relabeled by q' and q'', respectively." As a consequence of the forking instruction, the machine can be operating on multiple configurations, i.e. branches, in parallel and is inherently nondeterministic. The status of a machine at a given moment in a computation, called an *instantaneous description* (ID), is represented by the configurations that are present. Formally, an ID $\mathtt{u}$ is an element

$$\mathtt{C}_1 \vee \cdots \vee \mathtt{C}_m,$$

of the free commutative semigroup $(\mathtt{ID}(\mathtt{B}), \vee)$ generated by $\mathtt{Conf}(\mathtt{B})$.

Given a k-ACM $\mathtt{B} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P})$, let $\mathtt{P}_{\mathrm{com}} := \{xy \leq yx : x, y \in \mathtt{R}_k \cup \mathtt{Q}\}$. We note that $\mathtt{P}_{\mathrm{com}}$ is finite since $\mathtt{R}_k \cup \mathtt{Q}$ is finite. Hence $(X_\mathtt{B}, P_\mathtt{B})$ is an algebraic machine where $X_\mathtt{B} := \mathtt{R}_k \cup \mathtt{Q}$ and $P_\mathtt{B} := \mathtt{P} \cup \mathtt{P}_{\mathrm{com}}$. As before, will abuse the notation by using $\mathtt{B}$ to represent both structures. Let $\leq_\mathtt{B}$ be the computation relation for $\mathtt{B}$ and the compatible relations $\leq^p$ be given defined in Section 4.1.

Similar to Section 4.2, we define

$$\mathtt{Conf}(\mathtt{B}) := \{xqy \in X_{\mathtt{B}}^* : q \in \mathtt{Q}, xy \in \mathtt{R}_k^*\},$$

and $\mathtt{ID}(\mathtt{B}) := \mathtt{Conf}(\mathtt{B})^\vee$. The final term for $\mathtt{B}$ is the *final configuration* $\mathtt{C}_f := q_f$, and the set $\mathrm{Acc}(\mathtt{B})$ is as defined in Section 4.1.

Let $=_{com}$ be the equivalence relation generated by $\mathtt{P}_{\mathrm{com}}$. As in Equation (4.6), for all $\mathtt{C}, \mathtt{C}' \in \mathtt{Conf}(\mathtt{B})$, $\mathtt{C} =_{\mathrm{com}} \mathtt{C}'$ iff $\mathtt{C} =_{\mathrm{com}} q\mathtt{r}_1^{n_1} \cdots \mathtt{r}_k^{n_k} =_{\mathrm{com}} \mathtt{C}'$ for some $q \in \mathtt{Q}$ and $n_1, \ldots, n_k \in \mathbb{N}$. Since $t \in \mathtt{Conf}(\mathtt{B})$ iff t contains precisely one state variable, and each instruction $p : x \leq y$ is such that $x \in \mathtt{Conf}(\mathtt{B})$ and $y \in \mathtt{ID}(\mathtt{B})$, we immediately obtain:

Lemma 4.3.1. $\mathtt{Conf}(\mathtt{B})$ is stable for any k-ACM $\mathtt{B}$. In particular, $\mathtt{B}$ is $\mathtt{Q}$-stable.

For each $p \in \mathtt{P}$, we will abuse notation and write $\leq^p$ to mean $\leq^p_{\mathrm{com}}$.

4.3.1 Observations on admissibility. By definition, for all $s, t \in X_{\mathtt{B}}^*$, we obtain $st = \mathrm{com}ts$ and so $st \in \mathrm{Acc}(\mathtt{B})$ iff $ts \in \mathrm{Acc}(\mathtt{B})$. Therefore we obtain:

Lemma 4.3.2. Commutativity is strongly admissible in $\mathtt{B}$, for any k-ACM $\mathtt{B}$. Therefore $\mathbf{W}_{\mathtt{B}}^+ \in \mathsf{CRL}$.

By Corollary 4.1.10, and since all instructions in $P_{\mathtt{B}}$ are of type $\{\vee, \cdot, 1\}$, we deduce:

Lemma 4.3.3. Let $\mathtt{B}$ be a k-ACM. Then for any variety $\mathcal{V} \subseteq (\mathsf{C})\mathsf{RL}$ containing $\mathbf{W}_{\mathtt{B}}^+$, the complexity of the word problem, particularly the $\{\vee, \cdot, 1\}$-fragment, is at least as high as membership in $\mathrm{Acc}(\mathtt{B})$.

Given the admissibility of commutativity for any k-ACM $\mathtt{B} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P})$, henceforth we will implicitly assume $\mathbf{A}_X$, as defined in Section 4.1 is the free commutative semiring generated by $X = \mathtt{R}_k \cup \mathtt{Q}$.

4.3.2 Simulating CMs as ACMs and the Zero-Test Program. As demonstrated [17], considering only those computations that result in the final ID with all branches resulting in a final configuration q_F, i.e. where all registers are empty, is vital to the construction of our result. Such a convention allows us to implement a *program* that behaves like the zero-test instruction of a standard Counter Machine, i.e. a program that tests whether a given register is empty at a given state, and transitions to a new state only when the register is in fact empty. Such behavior cannot be directly implemented in a k-ACM, but can be simulated in its set of accepted IDs by augmenting its structure with the (sub)machine $\varnothing = (\mathtt{R}_k, \mathtt{Q}_\varnothing, \mathtt{P}_\varnothing)$, where $\mathtt{Q}_\varnothing = \{z_1, \ldots, z_k, q_F\}$ and set of instructions $\mathtt{P}_\varnothing$ are given by:

$$\begin{array}{rcrcl} \varnothing^i_j & : & z_i\mathtt{r}_j & \leq & z_i \\ \varnothing^i_F & : & z_i & \leq & q_F \vee q_F \end{array},$$

for each $i \in \{1, \ldots, k\}$ and $j \in \{1, \ldots, k\} \setminus \{i\}$.

We call the above machine the *zero-test program*, and we denote its computation relation by $\leq_\varnothing$. The zero-test program for a register $\mathtt{r}_i$ is implemented by a *zero-test* $\mathtt{r}_i$ *instruction* p, where p is of the form $q_{\text{in}} \leq q_{\text{out}} \vee z_i$. Since the desired final ID's of $\mathtt{B}_K$ consist only of joins of the configuration q_F, i.e. all registers are empty, the above instruction copies the contents of the registers and creates two paths; one path with the state q_{out} where $\mathtt{r}_i$ is intended to be empty, and the second with a state z_i where $\varnothing$ is intended to empty registers $\mathtt{r}_j$ and $\mathtt{r}_k$ and then output to the final state. Below is an example of implementing the zero-test on register $\mathtt{r}_1$ via the instruction $p : q_{\text{in}} \leq q_{\text{out}} \vee z_1$ on the configuration $q_{\text{in}}\mathtt{r}_1\mathtt{r}_2\mathtt{r}_3$:

$$\begin{array}{rcl} q_{\text{in}}\mathtt{r}_1\mathtt{r}_2\mathtt{r}_3 & \leq^p & q_{\text{out}}\mathtt{r}_1\mathtt{r}_2\mathtt{r}_3 \vee z_1\mathtt{r}_1\mathtt{r}_2\mathtt{r}_3 \\ & \leq^{\varnothing^1_2} & q_{\text{out}}\mathtt{r}_1\mathtt{r}_2\mathtt{r}_3 \vee z_1\mathtt{r}_1\mathtt{r}_3 \\ & \leq^{\varnothing^1_3} & q_{\text{out}}\mathtt{r}_1\mathtt{r}_2\mathtt{r}_3 \vee z_1\mathtt{r}_1 \\ & \leq^{\varnothing^1_F} & q_{\text{out}}\mathtt{r}_1\mathtt{r}_2\mathtt{r}_3 \vee q_F\mathtt{r}_1 \vee q_F\mathtt{r}_1. \end{array}$$

As we see, the above computation detected that register $\mathtt{r}_1$ is not empty in the configuration $q\mathtt{r}_1\mathtt{r}_2\mathtt{r}_3$ since the final ID contains the configuration $q_F\mathtt{r}_1$, and there are no q_F-instructions. In fact, $z_1\mathtt{r}_1\mathtt{r}_2\mathtt{r}_3 \notin \mathrm{Acc}(\emptyset)$ since there is no instruction applicable to the state z_1 which alters the contents of register $\mathtt{r}_1$. By a similar analysis, we obtain the following,

Proposition 4.3.4 ([17]). $z_i\mathtt{r}_1^{n_1}\cdots\mathtt{r}_k^{n_k} \in \mathrm{Acc}(\emptyset)$ if and only if $n_i = 0$.

Consequently, we obtain

Lemma 4.3.5. Let Σ be a set of simple equations. If Σ is not mingly then Σ is strongly-admissible for any k-ACM $\emptyset$-program.

Proof. This follows from Proposition 4.3.4, Lemma 4.1.14, and Lemma 4.3.1. □

For a given k-ACM $\mathtt{B} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P})$, we will call $\mathcal{P} = (\mathtt{R}_k, \mathtt{Q}_\mathcal{P}, \mathtt{P}_\mathcal{P})$ a *program* if $\mathtt{Q}_\mathcal{P} \leq \mathtt{Q}$ and $\mathtt{P}_\mathcal{P} \subseteq \mathtt{P}$, and by $\leq_\mathcal{P}$ we denote its corresponding computation relation. We define the relation $\sqsubseteq_\mathcal{P}$ on $\mathtt{Conf}(\mathtt{B})$ via $\mathtt{C} \sqsubseteq_\mathcal{P} \mathtt{D}$ iff $\mathtt{C} \leq_\mathcal{P} \mathtt{D}$ or $\mathtt{C} \leq_\mathcal{P} \mathtt{D} \vee u$ with $u \in \mathrm{ID}(\emptyset)$ and $u \in \mathrm{Acc}(\emptyset)$.[6] If $\mathcal{P}$ is a program containing no $\mathtt{Q}_\emptyset$-instructions, then $\mathtt{C} \sqsubseteq_\mathcal{P} \mathtt{D}$ iff there is a computation from $\mathtt{C}$ to $\mathtt{D} \vee u$ with instructions from $\mathcal{P} \cup \mathtt{P}_{\mathrm{com}}$ such that every zero-test was properly applied. Note that $\sqsubseteq_\mathcal{P}$ is transitive on configurations. All programs $\mathcal{P}$ defined henceforth will satisfy this property and, as a further consequence, if $\mathtt{C} \sqsubseteq_{\mathcal{P}\cup\emptyset} \mathtt{D}$, for some $\mathtt{D} \notin \mathtt{Conf}(\emptyset)$, then $\mathtt{C} \sqsubseteq_\mathcal{P} \mathtt{D}$.

Proposition 4.3.6. Let p be the instruction $q_{\mathrm{in}} \leq q_{\mathrm{out}} \vee z_i$ with distinct $q_{\mathrm{in}}, q_{\mathrm{out}} \notin \mathtt{Q}_\emptyset$. For $x, x' \in \mathtt{R}_3^*$, $q_{\mathrm{in}}x \sqsubseteq_{\{p\}} q_{\mathrm{out}}x'$ if and only if $x = x' = \mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3}$ and $n_i = 0$.

Proof. Let $x = \mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3}$. The only instruction applicable to $q_{\mathrm{in}}x$ is p, so

$$q_{\mathrm{in}} \leq q_{\mathrm{out}} \vee z_i \implies q_{\mathrm{in}}x \leq^p q_{\mathrm{out}}x \vee z_i x.$$

[6] If p is an instruction, by $\mathtt{C} \sqsubseteq_{\{p\}} \mathtt{D}$ we mean $\mathtt{C} \leq^p \mathtt{D} \vee u$.

Since the only instructions applicable are those from $\{p\}$ and $q_{\text{in}} \neq q_{\text{out}}$, the computation cannot proceed from this configuration. Hence,

$$q_{\text{in}}x \sqsubseteq_{\{p\}} q_{\text{out}}x' \iff x = x' \text{ and } z_i x \leq_\emptyset q_F \iff x = x' \text{ and } n_i = 0,$$

by Proposition 4.3.4. □

Proposition 4.3.7 ([17])**.** For every k-CM $\mathtt{M}$, there exists a k-ACM $\mathtt{B}$ such that for any configuration $\mathtt{C} \in \mathtt{Conf}(\mathtt{M})$, $\mathtt{C}$ is accepted in $\mathtt{B}$ iff $\theta(C) \in \mathrm{Acc}(\mathtt{B})$, for some map θ.

Proof. Let $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P})$ be a k-CM with final state q_f. We will suppose $\mathtt{Q}_\emptyset \setminus \{q_F\}$ and $\mathtt{Q}$ are disjoint and $q_F = q_f$. Consider the k-ACM $\mathtt{B} = (\mathtt{R}_k, \mathtt{Q}', \mathtt{P}')$, where $\mathtt{Q} := \mathtt{Q} \cup \mathtt{Q}_\emptyset$ with final state q_f, and $\mathtt{P}' := \mathtt{P} \cup \mathtt{P}_\emptyset$, with $\mathtt{P} := \theta[P]$ where

$$\begin{array}{rcl:rcl} \theta(q\mathtt{S}_i & \leq & q'\mathtt{S}_i\mathtt{r}_i) & q & \leq & q'\mathtt{r}_i \\ \theta(q\mathtt{S}_i\mathtt{r}_i & \leq & q'\mathtt{S}_i) & q\mathtt{r}_i & \leq & q' \\ \theta(q\mathtt{S}_i\mathtt{S}_{i+1} & \leq & q'\mathtt{S}_i\mathtt{S}_{i+1}) & q & \leq & q' \vee z_i, \end{array}$$

and for a configuration $\mathtt{C} = \langle q; n_1, \ldots, n_k \rangle$ of $\mathtt{M}$, $\theta(\mathtt{C}) := q\mathtt{r}_1^{n_1} \cdots \mathtt{r}_k^{n_k} \in \mathtt{Conf}(\mathtt{B}')$. We claim for every configuration $\mathtt{C}$ of $\mathtt{M}$, $\mathtt{C}$ is accepted by $\mathtt{M}$ if and only if $\theta(\mathtt{C}) \in \mathrm{Acc}(\mathtt{B}'$. Clearly, if $p \in \mathtt{P}$ is an increment or decrement instruction, then $\mathtt{C} \leq^p \mathtt{C}'$ if and only if $\theta(\mathtt{C}) \leq^{\theta(p)} \theta(\mathtt{C}')$. Furthermore, by Proposition 4.3.4, if p is a zero-test, then $\mathtt{C} \leq^p \mathtt{C}'$ if and only if $\theta(\mathtt{C}) \sqsubseteq_{\{\theta(p)\}} \theta(\mathtt{C}')$. Hence, for any configurations $\mathtt{C}, \mathtt{C}'$ of $\mathtt{M}$,

$$\mathtt{C} = \mathtt{C}_0 \leq^{p_1} \cdots \leq^{p_n} \mathtt{C}_n = \mathtt{C}' \iff \theta(\mathtt{C}) = \theta(\mathtt{C}_0) \sqsubseteq_{\theta(p_1)} \cdots \sqsubseteq_{\theta(p_n)} \theta(\mathtt{C}_n) = \theta(\mathtt{C}'),$$

for some configurations $\mathtt{C}_0, \ldots, \mathtt{C}_n$ in $\mathtt{M}$ and instructions $p_1, \ldots, p_n \in \mathtt{P}$. Hence $\mathtt{C}$ is accepted in $\mathtt{M}$ if and only if $\theta(\mathtt{C}) \in \mathrm{Acc}(\mathtt{B})$. □

Example 4.3.1. Consider the 1-CM $\mathtt{M}_{\mathrm{even}} = (\mathtt{R}_1, \mathtt{Q}_{\mathrm{even}}, \mathtt{P}_{\mathrm{even}})$. We simulate acceptance of $\mathtt{M}_{\mathrm{even}}$ by a 1-ACM $\mathtt{B}_{\mathrm{even}} = \mathtt{M}^{\vee}_{\mathrm{even}} := (\mathtt{R}_1, \mathtt{Q}^{\vee}_{\mathrm{even}}, \mathtt{P}^{\vee}_{\mathrm{even}})$ in the following way. Let $\mathtt{Q}^{\vee}_{\mathrm{even}} = \mathtt{Q}_{\mathrm{even}} \cup \{z_1\}$ and $\mathtt{P}_{\mathrm{even}}$ contains $\mathtt{P}_{\emptyset}$ and the instructions

$$\begin{array}{rcrcl} p_0 & : & q_0\mathtt{r}_1 & \leq & q_1 \\ p_1 & : & q_1\mathtt{r}_1 & \leq & q_0 \\ p_f & : & q_0 & \leq & q_f \vee z_1 \end{array}$$

By Proposition 4.3.7 and the machine $\tilde{\mathtt{M}}$ from Proposition 4.2.1, we obtain

Theorem 4.3.8. ([16, 20, 17]) There exists a 2-ACM $\tilde{\mathtt{B}} = (\mathtt{R}_2, \mathtt{Q}, \mathtt{P})$ such that membership of $\mathrm{Acc}(\tilde{\mathtt{B}})$ is undecidable.

Consequently, by Lemma 4.3.3 and Lemma 4.3.2

Theorem 4.3.9. The word problem is undecidable for CRL. More generally, for any variety $\mathcal{V} \subseteq \mathsf{RL}$ containing $\mathbf{W}^{+}_{\tilde{\mathtt{B}}}$, the word problem is undecidable, in particular for its $\{\vee, \cdot, 1\}$-fragment.

Corollary 4.3.10. Let $\mathcal{V}$ be an expansive subvariety of CRL containing $\mathbf{W}^{+}_{\mathtt{B}}$, for some k-ACM $\mathtt{B}$. Then the computational complexity of the equational theory of $\mathcal{V}$ is as high as the one of membership in $\mathrm{Acc}(\mathtt{B})$.

4.4 Non-primitive recursive lower bounds

In [23], Urquhart proves that there does not exist a *primitive recursive* decision procedure for $\mathbf{FL}_{\mathsf{ec}}$. Although $\mathbf{FL}_{\mathsf{ec}}$ has the FMP and is hence decidable, the proof establishing this fact is inherently nonconstructive. Urquhart actually proves that any decision procedure for $\mathbf{FL}_{\mathsf{ec}}$ is primitive recursive in the Ackermann function, which is a recursive function that is properly non-primitive recursive. Rendered algebraically, this proves that, while the equational theory of $\mathsf{CRL} + (x \leq x^2)$ is decidable, there is no primitive recursive decision procedure. That is, although it is decidable, there is no tractable procedure.

The purpose of this section is to note that the construction at the basis of Urquhart's proof exactly falls within the framework we have established above, and can therefore be naturally extended to capture complexity lower bounds for a more general class of simple equations. Although we provide a detailed outline of the main results, this section is intended to be supplemented by [23].

Before we provide the outline, some preliminary notions are needed. A more complete treatment can be found in the standard monograph [13].

A *space-bounded version* of the halting problem for off-line Turing machines is used as the intractable problem. An *off-line* Turing machine is defined to be a multi-tape Turing machine with a read-only input tape, a write-only output tape on which the head never moves left and a read-write work tape. Let Σ be a finite alphabet, and Σ^* the set of all finite strings over Σ, where $|\alpha|$ denotes the length of string $|\alpha|$. A machine M *accepts* $A \subseteq \Sigma^*$ if for all inputs $\alpha \in \Sigma^*$, M gives the output "1" iff $\alpha \in A$.

Let Σ_1, Σ_2 be finite alphabets. A function $f : \Sigma_1^* \to \Sigma_2^*$ *reduces* a set $A \subseteq \Sigma_1^*$ to a set $B \subseteq \Sigma_2^*$ provided that $\alpha \in A$ iff $f(\alpha) \in B$ for all $\alpha \in \Sigma^*$. If f reduces A to B, and in addition f is computable by a Turing machine that visits at most $\log_2(n)$ work tape squares during its computation on any word $\alpha \in \Sigma_1^*$ of length $n > 1$, then A is said to be *log-space reducible* to B; if in addition the length of $f(\alpha)$ is $O(|\alpha|)$, then A is *log-lin reducible* to B.

A set $A \subseteq \Sigma^*$ is said to be *decidable in space* $g : \mathbb{N} \to \mathbb{N}$ if there is a Turing machine that accepts A and visits at most $g(n)$ work tape squares during its computation on any word $\beta \in \Sigma^*$ of length n. The set A is *primitive recursive* if it is decidable in space g where g is a primitive recursive function. We note that the distinction between time and space is insignificant for the boundary between primitive recursive and non-primitive recursive algorithms. The lower bound for the space requirements of an algorithm imply a corresponding lower bound for time, since a machine must take at least one time step to visit a new square. The primitive recursive sets are closed under log-space reducibility in

the sense that if B is a primitive recursive set, and A is log-space reducible to B, then A is also primitive recursive.

4.4.1 An outline of the Urquhart construction. Essentially, Urquhart's construction establishes a complexity lower bound for membership of the quasi-equational theory for $\mathsf{CRL} + (x \leq x^2)$, and then uses a deduction theorem similar to Corollary 2.5.2 to establish the same for the equational theory.

The reduction of the intractable problem into $\mathsf{CRL} + (x \leq x^2)$ is given as follows:

1. Choose an enumeration of *off-line Turing machines* in which M_w is the machine encoded by the binary string w; where we assume that each machine occurs infinitely often in the enumeration. Define the set AHP (Ackermann-bounded version of the halting problem) to be

$$\{w : M_w \text{ accepts } w \text{ in space bounded by } A(|w|)\},$$

where A is a function, borrowed from [19], that majorizes all primitive recursive function. A is defined via $A(n) := A_n(2)$, where

$$\begin{aligned} A_0(x) &= 2x + 1, \\ A_{n+1}(x) &= A_n^{(x+1)}(0). \end{aligned} \tag{4.8}$$

2. Let $\mathtt{M}$ be a k-CM with an *initial state* q_1. Define the *initial configuration* $\mathtt{C}_1$ to be the one labeled by state q_1 with all registers empty, and we say the machine $\mathtt{M}$ *terminates* if $\mathtt{C}_1 \in \mathrm{Acc}(\mathtt{M})$. We say a computation of a machine is *bounded by* n if at every step in the computation, the contents of all the registers of the machine are bounded by n. The machine $\mathtt{M}$ is n-bounded if, when run from initial configuration, the resulting computation is bounded by n. Given that AHP is not primitive recursive (Thm. 4.2

[23]), a reduction from AHP to the set

$$\mathrm{ACP} := \{\mathtt{M} : \mathtt{M} = (\mathtt{R}_3, \mathtt{Q}, \mathtt{P}) \text{ is a terminating } A(|\mathtt{Q}|)\text{-bounded 3-CM}\},$$

establishes that ACP is not primitive recursive (Thm. 5.1 [23]).

3. Next, Urquhart defines a class of counter machines known as *expansive counter machines* (ECM). An ECM is a structure $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P} \cup \mathtt{P}_E)$, where $(\mathtt{R}_k, \mathtt{Q}, \mathtt{P})$ is a k-CM and $\mathtt{P}_E$ is a set of *expansive instructions* of the form $q\mathtt{S}_i\mathtt{r}_i \leq q\mathtt{S}_i\mathtt{r}_i^2$, for each $q \in \mathtt{Q}$ and register $\mathtt{r}_i$. Similarly, he defines the class of *expansive and-branching counter machines* (EACM), where an ECM is a structure $\mathtt{B} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P} \cup \mathtt{P}_E)$, where $(\mathtt{R}_k, \mathtt{Q}, \mathtt{P})$ is a k-ACM and $\mathtt{P}_E$ is a set of *expansive instructions* of the form $q\mathtt{r}_i \leq q\mathtt{r}_i^2$, for each $q \in \mathtt{Q}$ and register $\mathtt{r}_i$. *Termination* for these structures is defined analogously as above. In the same manner as Proposition 4.3.7, Urquhart shows that the set ECP is log-space reducible to EACP (Thm. 6.1 [23]), where

$$\begin{aligned} \mathrm{ECP} &:= \{\mathtt{M} : \mathtt{M} \text{ is a terminating ECM}\}, \\ \mathrm{EACP} &:= \{\mathtt{B} : \mathtt{B} \text{ is a terminating EACM}\}. \end{aligned}$$

4. The next step shows that ACP is log-lin reducible to ECP, which demonstrates that EACP is not primitive recursive. Let $\mathtt{M} = (\mathtt{R}_3, \mathtt{Q}, \mathtt{P})$ be a 3-CM and $n > 1$. Urquhart first constructs a k-CM $\mathtt{M}^{A(n)} = (\mathtt{R}_k, \mathtt{Q}^{A(n)}, \mathtt{P}^{A(n)})$ (where the value k happens to be $2n + 10$), such that

$$\mathtt{M} \text{ is } A(n)\text{-bounded} \iff \mathtt{C}_1^{A(n)} \in \mathrm{Acc}(\mathtt{M}^{A(n)}),$$

where $\mathtt{C}_1^{A(n)}$ is the initial configuration of $\mathtt{M}^{A(n)}$. Consider the corresponding ECM for $\mathtt{M}^{A(n)}$ given by $\mathtt{M}_E^{A(n)} = (\mathtt{R}_k, \mathtt{Q}^{A(n)}, \mathtt{P}^{A(n)} \cup \mathtt{P}_E)$. Urquhart then proves (Thm. 9.2)

$$\mathtt{C}_1^{A(n)} \in \mathrm{Acc}(\mathtt{M}^{A(n)}) \iff \mathtt{C}_1^{A(n)} \in \mathrm{Acc}(\mathtt{M}_E^{A(n)}),$$

where we note that $\mathtt{C}_1^{A(n)}$ is also the initial configuration of $\mathtt{M}_E^{A(n)}$. That is, $\mathtt{M}_E^{A(n)}$ terminates iff $\mathtt{C}_1^{A(n)} \in \mathrm{Acc}(\mathtt{M}_E^{A(n)})$ is witnessed by a computation with no expansive instructions iff $\mathtt{M}^{A(n)}$ terminates. Therefore we may conclude $\mathtt{M} = (\mathtt{R}_3, \mathtt{Q}, \mathtt{P})$ is $A(n)$-bounded iff $\mathtt{M}_E = (\mathtt{R}_3, \mathtt{Q}, \mathtt{P} \cup \mathtt{P}_E)$ is $A(n)$-bounded.

5. Lastly, using a deduction theorem (Thm. 7.1), essentially an $\mathbf{FL}$-rendering of Corollary 2.5.2, it is established that for any 3-EACM $\mathtt{M}$, $\mathtt{M}$ terminates iff $\mathbf{FL}_{\mathsf{ec}} \vdash \phi(\mathtt{M})$, where $\phi(\mathtt{M})$ is the formula encoding the instructions of $\mathtt{M}$ and the question of whether $\mathtt{M}$ terminates. Hence, AHP is log-space reducible to provability in $\mathbf{FL}_{\mathsf{ec}}$, and therefore $\mathbf{FL}_{\mathsf{ec}}$ has no primitive recursive decision procedure.

4.4.2 Observations of the construction. Our first observation begins with (3) from the above outline. The expansive instructions are meant to encode the effect of contraction $[\mathsf{c}] : x \leq x^2$ into an E(A)CM so that step (5) can be carried out. Clearly, for any expansive instruction of an E(A)CM $p : q\mathtt{S}_i\mathtt{r}_i \leq q\mathtt{S}_i\mathtt{r}_i^2$ ($q\mathtt{r}_i \leq q\mathtt{r}_i^2$),

$$t \leq^p t' \implies t \leq^{[\mathsf{c}]} t',$$

since p is an instance of contraction. Since $[\mathsf{c}]$ is not mingly, $[\mathsf{c}]$ is H-admissible in $\mathtt{M}$ by Lemma 4.1.14, for any $H \in \{\mathtt{Q}, \{\mathtt{S}_1\}, \ldots, \{\mathtt{S}_{k+1}\}\}$, where $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P})$ is a k-(A)CM. Hence instances of $[\mathsf{c}]$ can only be applied to terms in $\mathtt{R}_k^*$. Since the instance $q\mathtt{S}_i\mathtt{r}_i^n \leq q\mathtt{S}_i\mathtt{r}_i^{2n}$

can equally be obtained by the following computation:

$$q\mathtt{S}_i\mathtt{r}_i^n \leq^p q\mathtt{S}_i\mathtt{r}_i^{n+1} \leq^p \cdots \leq^p q\mathtt{S}_i\mathtt{r}_i^{2n-1} \leq^p q\mathtt{S}_i\mathtt{r}_i^{2n}, \tag{4.9}$$

we obtain

$$\mathtt{C} \in \mathrm{Acc}(\mathtt{cM}) \iff \mathtt{C} \in \mathrm{Acc}(\mathtt{M}_E), \tag{4.10}$$

where $\mathtt{M}_E = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P} \cup \mathtt{P}_E)$ is the corresponding E(A)CM for $\mathtt{M}$. That is, $[\mathtt{c}]$ is admissible in $\mathtt{M}_E$. Therefore, the word problem for $\mathsf{CRL} + [\mathtt{c}]$ is at least as complex as membership in $\mathrm{Acc}(\mathtt{M}_E)$ by Corollary 4.1.10.

For a k-(A)CM $\mathtt{M}$, we say $\mathtt{M}$ *is a terminating* $\mathtt{c}(\mathrm{A})\mathrm{CM}$ if $\mathtt{C}_1 \in \mathrm{Acc}(\mathtt{cM})$ where $\mathtt{C}_1$ is the initial configuration of $\mathtt{M}$. Similarly as in (3), we define

$$\begin{aligned} \mathtt{c}\mathrm{CP} &= \{\mathtt{M} : \mathtt{M} \text{ is a terminating } \mathtt{c}\mathrm{CM}\}, \\ \mathtt{c}\mathrm{ACP} &= \{\mathtt{M} : \mathtt{M} \text{ is a terminating } \mathtt{c}\mathrm{ACM}\}. \end{aligned} \tag{4.11}$$

The same argument (Thm. 6.1 [23]) establishes that $\mathtt{c}\mathrm{CP}$ is log-space reducible to $\mathtt{c}\mathrm{ACP}$. Using the machine notation from (4), by Equation (4.10) it follows that $\mathtt{M}$ is $A(n)$-bounded if and only if

$$\mathtt{C}_1^{A(n)} \in \mathrm{Acc}(\mathtt{M}^{A(n)}) \iff \mathtt{C}_1^{A(n)} \in \mathrm{Acc}(\mathtt{M}_E^{A(n)}) \iff \mathtt{C}_1^{A(n)} \in \mathrm{Acc}(\mathtt{cM}^{A(n)}).$$

Therefore, ACM is log-space reducible to $\mathtt{c}\mathrm{CP}$.

Define set of quasi-equations $\mathtt{c}\mathrm{QE}$ as follows: $\mathrm{acc}_{\mathtt{cM}}(\mathtt{C}_1) \in \mathtt{c}\mathrm{QE}$ if and only if $\mathtt{cM} = (\mathtt{R}_3, \mathtt{Q}, \mathtt{P} \cup \leq^{\mathtt{c}})$ is a terminating $A(|\mathtt{Q}|)$-bounded cACM with initial configuration $\mathtt{C}_1$. Observe that $\mathtt{c}\mathrm{ACP} = \{\mathtt{cM} : \mathrm{acc}_{\mathtt{cM}}(\mathtt{C}_1) \in \mathtt{c}\mathrm{QE}\}$.

By the reductions in the outline and the observations above, it is clear that AHP is log-space reducible to $\mathtt{c}\mathrm{QE}$. By Lemma 4.1.4, it follows that there is no primitive-recursive

decision procedure for the quasi-equational theory of $\mathsf{CRL} + [\mathsf{c}]$. Hence, as (5) in the outline, by Corollary 2.5.2 it follows that there is no primitive recursive decision procedure for the equational theory of $\mathsf{CRL} + [\mathsf{c}]$.

4.4.3 Weakly-expansive and expansive equations. We will now consider a class of single-variable equations for which the very same argument for the quasi-equational theory above can be carried out. We say a single-variable equation $x^{n_0} \leq x^{n_1} \vee \cdots \vee x^{n_m}$ is *weakly expansive* if $n_i > n_0$ for some $1 \leq i \leq m$, where $n_1, \ldots, n_m \geq 0$, $n_0 > 0$, and $m \geq 1$. Note that all expansive equations (as defined in Section 2.4) are weakly expansive. For illustrative purposes, we will consider the following weakly expansive equation

$$[\mathsf{d}] : x \leq x^2 \vee 1.$$

We claim that Urquhart's construction entails there is no primitive-recursive decision procedure for the quasi-equational theory of $\mathsf{CRL} + [\mathsf{d}]$. Define the sets $\mathsf{d}\mathrm{CP}$, $\mathsf{d}\mathrm{ACP}$, and $\mathsf{d}\mathrm{QE}$ as above by replacing $[\mathsf{c}]$ with $[\mathsf{d}]$. By the same argument (namely Thm. 6.1 [23]), $\mathsf{d}\mathrm{CP}$ is log-space reducible to $\mathsf{d}\mathrm{QE}$.

Therefore, since $\mathbf{W}^{+}_{\mathsf{dB}} \in \mathsf{CRL} + [\mathsf{d}]$ for any ACM $\mathtt{B}$, by Lemma 4.1.4 it follows that any decision procedure for the quasi-equational theory of $\mathsf{CRL} + [\mathsf{d}]$ is at least as complex as membership in $\mathsf{d}\mathrm{CP}$. To obtain the full result, we need only show that ACP is log-space reducible to $\mathsf{d}\mathrm{CP}$. We proceed with the machines in (4) from the outline. First observe

$$\mathtt{C}_1^{A(n)} \in \mathrm{Acc}(\mathtt{M}^{A(n)}) \implies \mathtt{C}_1^{A(n)} \in \mathrm{Acc}(\mathsf{d}\mathtt{M}^{A(n)}),$$

since all the proper instructions present in $\mathtt{M}^{A(n)}$ are also present in $\mathsf{d}\mathtt{M}^{A(n)}$ by definition. For the reverse direction, suppose $\mathtt{C}_1^{A(n)} \in \mathrm{Acc}(\mathsf{d}\mathtt{M}^{A(n)})$. Let $N \in \mathbb{N}$ be the smallest number for which

$$\mathtt{C}_1^{A(n)} \leq^{p_1} \mathtt{C}_2 \leq^{p_2} \cdots \mathtt{C}_N \leq^{\mathsf{d}} \mathtt{D} \vee \mathtt{D}' \in \mathrm{Acc}(\mathsf{d}\mathtt{M}^{A(n)}),$$

where $p_1, \ldots, p_{N-1} \in \mathtt{P}^{A(n)}$ are all proper instructions. If $N = 0$ then no $\leq^{\mathsf{d}}$ instruction are present in the computation, and hence $\mathtt{C}_1^{A(n)} \in \mathrm{Acc}(\mathtt{M}^{A(n)})$. Now suppose $N > 1$. We proceed by contradiction. By definition of $\mathtt{C}_N \leq^{\mathsf{d}} \mathtt{D} \vee \mathtt{D}'$, there exists monoid words x, u, v such that

$$\mathtt{C}_N = uxv \leq^{\mathsf{d}} ux^2v \vee uv = \mathtt{D} \vee \mathtt{D}'.$$

We note that x cannot contain any state-variable nor stopper variable by Lemma 4.1.14. Since $\mathtt{D} \vee \mathtt{D}' \in \mathrm{Acc}(\mathtt{M}^{A(n)})$, it follows that $ux^2v = \mathtt{D} \in \mathrm{Acc}(\mathtt{M}^{A(n)})$.

Now, by Equation (4.9), we obtain $uxv \leq_{\mathtt{M}_E^{A(n)}} ux^2v$. Hence

$$\mathtt{C}_1^{A(n)} \leq^{p_1} \cdots \mathtt{C}_N \leq_{\mathtt{M}_E^{A(n)}} \mathtt{D}.$$

Continuing in this way for each instance in the computation, i.e., selecting the "x^2" branch of each instance of $[\mathsf{d}]$, we recover the computation in $\mathtt{M}_E^{A(n)}$ witnessing $\mathtt{C}_1^{A(n)} \in \mathrm{Acc}(\mathtt{M}_E^{A(n)})$. This contradicts step (4) in Urquhart's construction since $\mathtt{C}_1^{A(n)} \in \mathrm{Acc}(\mathtt{M}_E^{A(n)})$ only if no expansive instructions are present in the computation. Therefore $N \not> 1$ and we are done.

Proposition 4.4.1. The quasi-equational theory of $\mathsf{CRL} + [x \leq x^2 \vee 1]$ has no primitive recursive decision procedure.

We note that the only requirements that make the above argument succeed are that: (i) the equation is single-variable and not mingly to ensure $\mathrm{Acc}(\mathtt{M}) \subseteq \mathtt{Conf}(\mathtt{M})^{\vee}$, and (ii) the equation is weakly expansive to ensure at least one branch of an instance of $[\mathsf{d}]$ can be obtained by a sequence of expansive instructions. Let $[\mathrm{R}] = (\mathbf{1_{n_0}}, \mathrm{R})$ be the simple equation obtained by linearizing a weakly expansive equation $x^{n_0} \leq x^{n_1} \vee \cdots \vee x^{n_m}$, and without loss of generality suppose $n_1 > n_0$. By linearization, for each $1 \leq i \leq n_0$

$$x_1 \cdots x_{i-1} \cdot x_i^c \cdot x_{i+1} \cdots x_{n_0} \in \mathrm{R}, \tag{4.12}$$

where $c := n_1 - n_0 > 0$. It is immediate that (i) and (ii) hold for $[\mathrm{R}]$.

Corollary 4.4.2. Let $[\mathrm{R}]$ be the simple equation of some single-variable weakly expansive equation. Then there is no primitive recursive decision procedure for the quasi-equational theory of $\mathsf{CRL} + [\mathrm{R}]$. If $[\mathrm{R}]$ was obtained from an expansive equation, then there is no primitive recursive decision procedure for the equational theory of $\mathsf{CRL} + [\mathrm{R}]$.

In particular, let $[\mathsf{k}_n^m] : x^n \leq x^{n+m}$, for $m, n \geq 1$ be an expansive knotted rule. Urquhart's result shows that, even though it is decidable by Proposition 3.1.1, the equational theory $\mathsf{CRL} + [\mathsf{k}_n^m]$ does not have a primitive recursive decision procedure.

Furthermore, we note that for any weakly expansive equation $[\mathsf{d}] : x^n \leq x^{m+n} \vee 1$, where $m, n \geq 1$, there is no primitive recursive decision procedure for the quasi-equational theory of $\mathsf{CRL} + [\mathsf{d}]$.

Chapter 5: Undecidability and the class $\mathcal{U}$ of simple equations

This chapter is devoted to establishing new undecidability results, particularly for extensions of CRL by simple equations, utilizing the techniques developed in the previous chapter. Both the first and second sections proceed along similar lines, the former focusing on the $\{\vee, \cdot, 1\}$-fragment for extensions of $(\mathsf{C})\mathsf{RL}$, and the latter focusing on $\{\leq, \cdot, 1\}$-fragment for extensions of RL. Specifically we prove undecidability results for the extension by equations from a class $\mathcal{U}$ of simple equations. For CRL, Theorem 5.3.1 establishes that the word problem for extensions from $\mathcal{U}$ is undecidable. Consequently, using the deduction theorem from Section 2.5, our capstone Theorem 5.1.13 proves that the equational theory for $\mathsf{CRL} + [\mathrm{D}]$ is undecidable for any expansive $[\mathrm{D}] \in \mathcal{U}$. Equivalently, this shows that provability in the corresponding substructural logic $\mathbf{FL}_{\mathsf{e}} + (\mathrm{D})$ is undecidable. For example, the equation $[\mathrm{D}] : x \leq x^2 \vee x^3$ is an expansive member of $\mathcal{U}$, so the equational theory of $\mathsf{CRL} + [\mathrm{D}]$ is undecidable, and therefore provability is undecidable in $\mathsf{FL}_{\mathsf{e}} + (\mathrm{D})$ where (D) is the structural rule

$$\frac{\Delta_1, \Gamma, \Gamma, \Delta_2 \Rightarrow \Pi \quad \Delta_1, \Gamma, \Gamma, \Gamma, \Delta_2 \Rightarrow \Pi}{\Delta_1, \Gamma, \Delta_2 \Rightarrow \Pi}\ (\mathrm{D}).$$

In the last section, we provide a characterization for the class of equations $\mathcal{U}$ which is essential for both Theorem 5.3.1 and Theorem 5.2.6. The definition of $\mathcal{U}$ is equivalently stated via, $[\mathrm{D}] \in \mathcal{U}$ if and only if $\mathsf{CRL} + [\mathrm{D}] \not\models [V]$, for some *spinal equation* $[V]$ of the form:

$$[V] : x_1^{f(1)} \cdots x_k^{f(k)} \leq 1 \ \vee \ x_1^{v_1(1)} \ \vee \ x_1^{v_2(1)} x_2^{v_2(2)} \ \vee \cdots \vee \ x_1^{v_k(1)} \cdots x_k^{v_k(k)},$$

for some $k \geq 1$ and vectors $f, v_1, \ldots, v_k \in \mathbb{N}^k$ such that $f \neq v_k$ and $v_i(i) > 0$ for each $i = 1, \ldots, k$. The goal of this section is to establish that such non-spinal equations satisfy a condition that guarantees admissibility for the machines defined in Section 5.1 and 5.2. However, the techniques needed to prove this claim are quite distinct and unrelated to those needed in rest of the chapter, which is why they are presented last. We show that the property of satisfying a spinal equation is related to whether or not there exists positive solutions to some corresponding systems of linear equations in $\mathbb{R}^n$. Each joinand of an equation will be associated to some vector, and the right-hand side of simple equations as a set of vectors, which we may view as a matrix. In this context, monoid substitutions will also correspond to an associated matrix, and applications of a substitution as the transformation, or product, by this matrix. In this way, a simple equation is a member of $\mathcal{U}$ if and only if its associated matrix does not appear in the decomposition of some spinal equation in terms of *upper-triangular block matrices*. Furthermore, we show that this is equivalent to satisfying the sufficient condition of admissibility defined in Section 5.1.3.

5.1 Admissibility for ACMs

Our goal will be to find proper subvarieties of $(\mathsf{C})\mathsf{RL}$ for which Corollary 4.1.6 will be applicable, as well as strengthening this result to the equational theory for some expansive subvarieties of CRL using the deduction theorem Corollary 2.5.2. Motivated by Chapter 4, we will restrict our attention to varieties axiomatized by equations in the signature $\{\vee, \cdot, 1\}$, i.e., ISR-axioms. We will further restrict our attention to only simple equations, since degenerate equations correspond to 1-element models in RL and non-degenerate non-proper equations are RL-equivalent to simple equations (see Proposition 2.1.2).

5.1.1 Motivation for axiomatic extensions of CRL. Consider the 1-ACM $\mathtt{B}_{\mathrm{even}}$ from Example 4.2.1 and note that its computations faithfully represent the inequality relation in CRL. If we consider the inequality relation in $\mathsf{CRL} + [\mathrm{D}]$, where $[\mathrm{D}]$ is the equation $(\forall x)\ x \leq x^2 \vee x^4$, we observe that for a machine to faithfully represent the associated

inequality relation it must further admit the "ambient instruction" given by

$$t \leq^{\mathrm{D}} t^2 \vee t^4,$$

for all $t \in (\mathtt{Q}_{\mathrm{even}} \cup \mathtt{R}_1)^*$ in addition to being closed under the inference rules $[\cdot]$ and $[\vee]$. Let $\leq_{\mathrm{DB}_{\mathrm{even}}}$ be the smallest compatible preorder generated by $\mathtt{P}_{\mathrm{even}} \cup \leq^{\mathrm{D}}$, and define $\mathrm{Acc}(\mathrm{DB}_{\mathrm{even}})$ be the set of accepted ID's under the relation $\leq_{\mathrm{D}(\mathtt{B})}$. Clearly, $\mathrm{Acc}(\mathtt{B}_{\mathrm{even}}) \subseteq \mathrm{Acc}(\mathrm{DB}_{\mathrm{even}})$, and since there are no instructions (nor instances of $\leq^{\mathrm{D}}$) that remove state variables we obtain $\mathrm{Acc}(\mathrm{DB}_{\mathrm{even}}) \subseteq \mathtt{ID}(\mathtt{B}_{\mathrm{even}})$. However, $q_0\mathtt{r}_1^3 \notin \mathrm{Acc}(\mathtt{B}_{\mathrm{even}})$, but $q_0\mathtt{r}_1^3 \in \mathrm{Acc}(\mathrm{DB}_{\mathrm{even}})$ since

$$q_0\mathtt{r}_1^3 \leq^{\mathrm{D}} q_0\mathtt{r}_1^6 \vee q_0\mathtt{r}_1^{12} \in \mathrm{Acc}(\mathtt{B}_{\mathrm{even}}).$$

It is clear that the expansion of the machine by the ambient instruction (needed for representing the inequality relation in $\mathsf{CRL} + [\mathrm{D}]$) does not have the same computation relation, or put differently, the machine $\mathtt{B}_{\mathrm{even}}$ is not suitable for representing the inequality relation in $\mathsf{CRL} + [\mathrm{D}]$ because these ambient instructions are not already *admissible* in it. Likewise, there is no guarantee that there is a machine that has an undecidable acceptance problem (for example the machine $\tilde{\mathtt{B}}$) and in which these ambient instructions are available/admissible. For that reason we cannot use the same argumentation to show that $\mathsf{CRL} + [\mathrm{D}]$ has undecidable word problem.

Exactly the same issue occurs if the simple equation is contraction $x \leq x^2$. Actually, for the case of contraction not only does this particular encoding fail to be faithful, but there is no faithful encoding of an undecidable machine: the word problem for $\mathsf{CRL} + [\mathsf{c}]$ is actually decidable. We will show that even though for the equation $[\mathrm{D}]$ above our current encoding is problematic (as is with contraction), surprisingly, unlike with contraction, there is a different encoding that works for $[\mathrm{D}]$; this will allow us to prove that the word problem for $\mathsf{CRL} + [\mathrm{D}]$ is undecidable. We present the idea of this new encoding by showing that it

faithfully encodes the machine $\mathtt{B}_{\text{even}}$. We will actually see that what makes it work is that the new encoding is such that even if the ambient instructions were available, they would not contribute to any increase in the accepted configurations; this is a rephrasing of what we referred to as: the given equation is *admissible* in the particular machine.

The idea is to construct a new machine $\mathtt{B}_K$, for an appropriate integer K, as a modification of $\mathtt{B}_{\text{even}}$ that manages to replace the decrement instructions p_0, p_1 by programs $\mathcal{P}_0, \mathcal{P}_1$, respectively, that divide the contents of register $\mathtt{r}_1$ by a fixed constant K; for example $q_0\mathtt{r}_1^{K\ell} \leq_{\mathcal{P}_0} q_1\mathtt{r}_1^{\ell}$, and more specifically, $q_0\mathtt{r}_1^{K^{n+1}} \leq_{\mathcal{P}_0} q_1\mathtt{r}_1^{K^n}$. In this case, we will say a term is accepted if it computes a join of configurations of the form $q_f\mathtt{r}_1$, so $q_0\mathtt{r}_1^n \in \mathrm{Acc}(\mathtt{B}_K)$ iff $n = K^{2m}$ for some $m \geq 0$. That is, we put a necessary condition on configurations to be accepted.[1] For our equation $[\mathrm{D}]$, if we set $K \geq 3$, it is easy to verify that if

$$q\mathtt{r}_1^n\mathtt{r}_1^m \leq^{\mathrm{D}} q\mathtt{r}_1^n\mathtt{r}_1^{2m} \vee q\mathtt{r}_1^n\mathtt{r}_1^{4m} = q\mathtt{r}_1^{n+2m} \vee q\mathtt{r}_1^{n+4m},$$

the only way $n + 2m$ and $n + 4m$ are both powers of K is if $m = 0$, and hence an instance of $[\mathrm{D}]$ in a computation, at least with respect to being accepted, is superfluous.[2] Thus we obtain

$$q\mathtt{r}_1^n\mathtt{r}_1^{2m} \vee q\mathtt{r}_1^n\mathtt{r}_1^{4m} \in \mathrm{Acc}(\mathtt{B}_K) \implies q\mathtt{r}_1^n\mathtt{r}_1^m \in \mathrm{Acc}(\mathtt{B}_K),$$

and thus $\mathrm{Acc}(\mathtt{B}_K) = \mathrm{Acc}(\mathrm{D}\mathtt{B}_K)$. So, the equation $[\mathrm{D}]$ is *admissible* in the machine $\mathtt{B}_K$.

In the next sections we will make rigorous the notions of admissibility, the machines $\mathtt{B}_K$, and a class of simple equations $\mathcal{U}$ that are admissible for such ACMs. This will prove, in particular, that $\mathsf{CRL} + [\mathrm{D}]$ has an undecidable word problem for any $[\mathrm{D}] \in \mathcal{U}$.

---

[1] This definition of acceptance for the machine $\mathtt{B}'$ is for heuristic convenience. In Section 5.1.2, to properly define programs to multiply/divide by K, we will need to add a fresh variable q_F, acting as a new final state, and a set of instructions that put $q_f\mathtt{r}_1 \leq_{\mathtt{B}_K} q_F$.

[2] If $n + 2m = K^a$ and $n + 4m = K^{a+b}$, for some $a \geq 0$ and $b \geq 1$, then $K^a \geq 2m = K^{a+b} - K^a \geq K^a(K - 1)$, and hence $K \leq 2$.

5.1.2 The $\mathtt{B}_K$ Machine. Given any 2-ACM $\mathtt{B} = (\mathtt{R}_2, \mathtt{Q}, \mathtt{P})$ and simple equation $[\mathrm{D}]$, our ultimate goal is to construct a new machine $\mathtt{B}'$ that *simulates* the machine $\mathtt{B}$ such that $[\mathrm{D}]$ is strongly admissible in $\mathtt{B}'$, i.e. $\mathrm{Acc}(\mathrm{DB}') = \mathrm{Acc}(\mathtt{B}')$. This can be achieved for all $[\mathrm{D}] \in \mathcal{U}$ by constructing a 3-ACM $\mathtt{B}_K = (\mathtt{R}_3, \mathtt{Q}_K, \mathtt{P}_K)$, for some $K > 1$ provided by Lemma 5.3.13, that will simulate the acceptance of the 2-ACM $\mathtt{B}$ in the following way:

$$\mathtt{C} \in \mathrm{Acc}(\mathtt{B}) \text{ if and only if } \mathtt{C}_K \in \mathrm{Acc}(\mathtt{B}_K),$$

where $\mathtt{C} \in \mathtt{Conf}(\mathtt{B})$ and $(q\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2})_K := q\mathtt{r}_1^{K^{n_1}}\mathtt{r}_2^{K^{n_2}}$. We will have $\mathtt{Q} \subset \mathtt{Q}_K$, a new final state $q_F \in \mathtt{Q}_K \setminus \mathtt{Q}$, and for the instructions $\mathtt{P}_K$, replacing each increment and decrement instruction of $\mathtt{B}$ by the programs *multiply by* K and *divide by* K, respectively, with the corresponding pair of states, while keeping all forking instructions of $\mathtt{B}$ the same.

We recall the machine $\emptyset = (\mathtt{R}_3, \mathtt{Q}_0, \mathtt{P}_\emptyset)$ defined in Section 4.3.2. The zero-test instructions $q \leq q' \vee z_i$ will be used to define the multiply and division programs.[3] We will construct these programs from simpler programs named *transfer*, *add-*K, and *subtract-*K. We will assume that all state names defined by the following machines are disjoint from each other, disjoint from $\mathtt{Q}_\emptyset$, and disjoint from the set of states $\mathtt{Q}$ from a fixed 2-ACM $\mathtt{B}$.

A *transfer program* $\mathtt{T}_i(q_{\text{out}}) = (\mathtt{R}_3, \mathtt{Q}_{\mathtt{T}_i}, \mathtt{P}_{\mathtt{T}_i}(q_{\text{out}}))$ is meant to transfer all contents in register $\mathtt{r}_3$ to register $\mathtt{r}_i$ and output state q_{out}. We define the set $\mathtt{Q}_{\mathtt{T}_i} = \{t_0, t_1\}$ and the set of instructions $\mathtt{P}_{\mathtt{T}_i}(q_{\text{out}}) = \{T_-, T_+, T_{\text{out}}\}$, where:

$$\begin{array}{lcrcl} T_- & : & t_0\mathtt{r}_3 & \leq & t_1 \\ T_+ & : & t_1 & \leq & t_0\mathtt{r}_i \\ T_{\text{out}} & : & t_0 & \leq & q_{\text{out}} \vee z_3 \end{array} .$$

[3] As we will see later, this same construction can be implemented for CMs using the instructions $q\ 0\mathtt{r}_i\ q'$.

Below is an example of $\mathtt{T}_1(q_{\text{out}})$ running on the configuration $t_0\mathtt{r}_3^2$:

$$t_0\mathtt{r}_3^2 \leq^{T_-} t_1\mathtt{r}_3 \leq^{T_+} t_0\mathtt{r}_1\mathtt{r}_3 \leq^{T_-} t_1\mathtt{r}_1 \leq^{T_+} t_0\mathtt{r}_1^2 \sqsubseteq_{\{T_{\text{out}}\}} q_{\text{out}}\mathtt{r}_1^2.$$

We define $\leq_{\mathtt{T}_i}$ to be the computation relation of the transfer program $\mathtt{T}_i(q_{\text{out}})$.

Proposition 5.1.1. Let $\mathtt{T}_i(q_{\text{out}})$ be a transfer program with $\{i,j\} = \{1,2\}$. If $\delta \in \{0,1\}$, then $t_\delta\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\mathtt{T}_i} q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$ if and only if $m_3 = 0$, $m_i = n_i + n_3 + \delta$, and $m_j = n_j$.

Proof. Without loss of generality, suppose $i = 1$. We proceed by induction on n_3. Since the only instruction applicable to a configuration labeled by state t_1 is T_+, we will first examine only the case that $\delta = 0$. If $n_3 = 0$, then the only instruction applicable to $t_0\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}$ is T_{out}, thus

$$t_0\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2} \leq^{T_{\text{out}}} q_{\text{out}}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2} \vee z_3\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2},$$

and $z_3\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2} \in \text{Acc}(\emptyset)$ since $n_3 = 0$. Hence

$$t_0\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\mathtt{T}_1} q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$$

if and only if $m_3 = 0$, $m_i = n_i + n_3$, and $m_j = n_j$.

Now suppose the claim holds for some $n_3 \geq 0$. The only instructions applicable to $t_0\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+1}$ are T_- and T_{out}. On the one hand,

$$t_0\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+1} \leq^{T_{\text{out}}} q_{\text{out}}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2} \vee z_3\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+1},$$

but by Proposition 4.3.6, $z_3\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+1} \notin \text{Acc}(\emptyset)$. So the only possible instruction applicable is T_-, hence

$$t_0\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+1} \leq^{T_-} t_1\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3}.$$

Now,

$$t_1\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \leq^{T_+} t_0\mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\mathtt{T}_1} q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$$

if and only if $m_1 = n_1 + n_3 + 1$, $m_2 = n_2$, and $m_3 = 0$, by the induction hypothesis. Hence

$$t_0\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+1} \sqsubseteq_{\mathtt{T}_1} q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$$

if and only if $m_1 = n_1 + n_3 + 1 = n_1 + (n_3 + 1)$, $m_2 = n_2$, and $m_3 = 0$, completing the induction. Hence, for $\delta = 1$, since the only applicable instruction to $t_1\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3}$ is T_+,

$$t_1\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \leq^{T_+} t_0\mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\mathtt{T}_1} q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$$

where the latter $\sqsubseteq_{\mathtt{T}_1}$ is achieved iff $m_1 = (n_1 + 1) + n_3 = n_1 + n_3 + 1$, $m_2 = n_2$, and $m_3 = 0$ by the above argument. □

The *add-K program* is denoted by $+K = (\mathtt{R}_3, \mathtt{Q}_{+K}, \mathtt{P}_{+K})$, and is intended to add K tokens to register $\mathtt{r}_3$ and output with state a_K. We define the set $\mathtt{Q}_{+K} = \{a_0, \ldots, a_K\}$, and the set of instructions $\mathtt{P}_{+K} = \{+_1, \ldots, +_K\}$, where:

$$\begin{array}{lcrcl} +_1 & : & a_0 & \leq & a_1\mathtt{r}_3 \\ +_2 & : & a_1 & \leq & a_2\mathtt{r}_3 \\ \vdots & & & \vdots & \\ +_K & : & a_{K-1} & \leq & a_K\mathtt{r}_3 \end{array} .$$

We define $\leq_+$ to be the computation relation on the add-K program. Note that the above program is deterministic on a single configuration, and it is easily verified that $a_0 \leq_+ a_K\mathtt{r}_i^K$. In fact,

Proposition 5.1.2. Let $0 \leq \delta \leq K$ and u be an ID. Then $a_\delta \mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \leq_+ u$ if and only if $u = q\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$ where $q = a_{\delta'}$ for $\delta \leq \delta' \leq K$, $m_3 = n_3 + (\delta' - \delta)$, $m_1 = n_1$ and $m_2 = n_3$.

We now define the *multiply by* K programs, denoted by $\times_i(q_{\text{in}}, q_{\text{out}}) = (\mathtt{R}_3, \mathtt{Q}_{\times_i}, \mathtt{P}_{\times_i})$, where $i \in \{1, 2\}$. This program is meant to multiply the contents of $\mathtt{r}_i$ by K, with input state q_{in} and output state q_{out}. We define the set $\mathtt{Q}_\times = \mathtt{Q}_{+K} \cup \mathtt{Q}_{\mathtt{T}_i}$ and the set $\mathtt{P}_{\times_i} = \mathtt{P}_{\mathtt{T}_i}(q_{\text{out}}) \cup \mathtt{P}_{+K} \cup \{\times_{\text{in}}, \times_{\text{loop}}, \times_{\text{out}}\}$, where:

$$\begin{array}{lcrcl} \times_{\text{in}} & : & q_{\text{in}} & \leq & a_K \vee z_3 \\ \times_{\text{loop}} & : & a_K\mathtt{r}_i & \leq & a_0 \\ \times_{\text{out}} & : & a_K & \leq & t_0 \vee z_i \end{array} ,$$

where the initial instruction $\times_{\text{in}}$ is meant to verify that register $\mathtt{r}_3$ is empty and initiate the multiplication process. That is, a token in register $\mathtt{r}_i$ is removed and K tokens are added to $\mathtt{r}_3$ by the instruction $\times_{\text{loop}}$ and the program $+K$ repeatedly until all tokens are removed from $\mathtt{r}_i$. Once $\mathtt{r}_i$ is emptied, $\times_3$ transfers the tokens in $\mathtt{r}_3$ to $\mathtt{r}_i$.

Below is an example of $\times_1(q_{\text{in}}, q_{\text{out}})$ running on the configuration $q_{\text{in}}\mathtt{r}_1^2\mathtt{r}_2$:

$$\begin{array}{lll} q_{\text{in}}\mathtt{r}_1^2\mathtt{r}_2 & \sqsubseteq_{\{\times_{\text{in}}\}} & a_K\mathtt{r}_1^2\mathtt{r}_2 \\ & \leq^{\times_{\text{loop}}} & a_0\mathtt{r}_1\mathtt{r}_2 \\ & \sqsubseteq_+ & a_K\mathtt{r}_1\mathtt{r}_2\mathtt{r}_3^K \\ & \leq^{\times_{\text{loop}}} & a_0\mathtt{r}_2\mathtt{r}_3^K \\ & \sqsubseteq_+ & a_K\mathtt{r}_2\mathtt{r}_3^{2K} \\ & \sqsubseteq_{\{\times_{\text{out}}\}} & t_0\mathtt{r}_2\mathtt{r}_3^{2K} \\ & \sqsubseteq_{\mathtt{T}_1} & q_{\text{out}}\mathtt{r}_1^{2K}\mathtt{r}_2. \end{array}$$

We define $\leq_{\times_i(q_{\text{in}},q_{\text{out}})}$ to be the computation relation on the multiply by K program, and will write $\leq_{\times_i}$ when the program is understood in context. We are interested in the consequences of a single run of a $\times_i(q_{\text{in}}, q_{\text{out}})$ program starting from a given configuration. This requires special care in the case that $q_{\text{in}} = q_{\text{out}}$, but is characterized by how many times the instruction $\times_{\text{in}}$ is implemented in a computation.

Proposition 5.1.3. Let $0 \leq \delta \leq K$. Then $a_\delta \mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\times_i} q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$, witnessed by a computation with no instance of instruction $\times_{\text{in}}$, if and only if $m_j = n_j$, $m_3 = 0$, and $m_i = Kn_i + n_3 + (K - \delta)$. Hence $q_{\text{in}}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\times_i} q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$, witnessed by a computation with precisely one instance of $\times_{\text{in}}$, if and only if $m_j = n_j$, $m_3 = n_3 = 0$, and $m_i = Kn_i$.

Proof. Note that the only instruction that outputs the state q_{out} is $T_{\text{out}} \in \mathtt{P}_{\mathtt{T}_i}$, which is only applicable to an ID containing a configuration labeled by state t_0 in the subprogram $\mathtt{T}_i$. Now, the only instruction in $\mathtt{P}_{\times_i}$ that outputs a state in $\mathtt{Q}_{\mathtt{T}_i}$ is $\times_{\text{out}}$ which is only applicable to an ID containing a configuration labeled by state a_K. Since the only instructions applicable to a state $a_{\delta'}$, for $0 \leq \delta' < K$, are those from $\mathtt{P}_{+K}$, we obtain $a_\delta x \sqsubseteq_{\times_i} q_{\text{out}}x'$ iff there is a computation

$$a_\delta x \leq_{+K} a_K x_1 \sqsubseteq_{\times_i} t_0 x_2 \sqsubseteq_{\mathtt{T}_i} q_{\text{out}}x',$$

for some $x, x', x_1, x_2 \in \mathtt{R}_3^*$. We will prove the above claim only for $i = 1$, since the proof for the other case is identical. We proceed by induction on n_1. For $n_1 = 0$, observe that

$$\begin{array}{llll}
a_\delta \mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} & \leq_{+K} \; a_K x & \iff & x = \mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+(K-\delta)} \quad \text{by Prop. 5.1.2,} \\
a_K \mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+(K-\delta)} & \sqsubseteq_{\times_1} \; t_0 x & \iff & x = \mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+(K-\delta)} \quad \text{by Prop. 4.3.6,} \\
t_0 \mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+(K-\delta)} & \sqsubseteq_{\mathtt{T}_1} \; q_{\text{out}} x & \iff & x = \mathtt{r}_1^{n_3+(K-\delta)}\mathtt{r}_2^{n_2} \quad \text{by Prop. 5.1.1.}
\end{array}$$

Note that there are no instructions applicable to a configuration with state q_{out} except $\times_{\text{in}}$ in the case that $q_{\text{in}} = q_{\text{out}}$. Hence, by the observation above, $a_\delta \mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_\times q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$

if and only if $m_2 = n_2$, $m_3 = 0$, and $m_1 = n_3 + (K - \delta)$. Now suppose the claim holds for some $n_1 \geq 0$. Observe that

$$a_\delta \mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \leq_{+K} a_K x \iff x = \mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+(K-\delta)} \text{ by Prop. 5.1.2.}$$

Now, the only instructions applicable to $a_K \mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+(K-\delta)}$ are $\times_{\text{loop}}$ and $\times_{\text{out}}$. We see that the latter must be excluded since

$$a_K \mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+(K-\delta)} \leq^{\times_{\text{out}}} (t_0 \vee z_1)\mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+(K-\delta)},$$

but $z_1 \mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+(K-\delta)} \notin \mathrm{Acc}(\emptyset)$ by Proposition 4.3.4. Hence the only instruction that allows the computation to proceed is $\times_{\text{loop}}$, and thus we obtain

$$a_K \mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+(K-\delta)} \sqsubseteq_{\{\times_{\text{loop}}, \times_{\text{out}}\}} qx$$

iff $x = \mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+(K-\delta)}$ with $q = a_0$, and

$$a_0 \mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+(K-\delta)} \sqsubseteq_{\times_1} q_{\text{out}} x$$

iff $x = \mathtt{r}_1^{Kn_1+n_3+(K-\delta)}\mathtt{r}_2^{n_2}$ by the induction hypothesis. Thus our claim is satisfied.

Now, the only instruction applicable to $q_{\text{in}}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3}$ is the zero-test $\times_{\text{in}}$. So $q_{\text{in}} x \sqsubseteq_\times q_{\text{out}} x'$ only if $n_3 = 0$. Thus we obtain,

$$q_{\text{in}}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\{\times_{\text{in}}\}} a_K x \iff x = \mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \text{ and } n_3 = 0,$$

where $a_K \mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_\times q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$ if and only if $m_2 = n_2$, $m_3 = n_3 = 0$, and $m_i = Kn_i$ by the above. □

Next we define the *subtract-K program*, denoted by $-_iK = (\mathtt{R}_3, \mathtt{Q}_{-_iK}, \mathtt{P}_{-_iK})$, for $i \in \{1, 2\}$, which is meant to subtract K tokens to register $\mathtt{r}_i$ and output state s_K. We define $\mathtt{Q}_{-_i} = \{s_0, \ldots, s_K\}$ and the instructions $\mathtt{P}_{-_iK} = \{-_1, \ldots, -_K\}$, where:

$$\begin{array}{lcrcl} -_1 & : & s_0\mathtt{r}_i & \leq & s_1 \\ -_2 & : & s_1\mathtt{r}_i & \leq & s_2 \\ \vdots & & & \vdots & \\ -_K & : & s_{K-1}\mathtt{r}_i & \leq & s_K \end{array} .$$

We denote the computation relation on the subtract-K program by $\leq_-$. It is easy to see that the subtract-K program is deterministic on a single configuration. Furthermore, it easily follows that:

Proposition 5.1.4. Let $\{i, j\} = \{1, 2\}$. If $0 \leq \delta \leq K$, then $s_\delta \mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_- s_K\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$ if and only if $n_i \geq K - \delta$, $m_i = n_i - (K - \delta)$, and $m_j = n_j$.

We are now ready to define the *divide by K* program for $i \in \{1, 2\}$, denoted by $\div_i(q_{\text{in}}, q_{\text{out}}) = (\mathtt{R}_3, \mathtt{Q}_{\div_i}, \mathtt{P}_{\div_i}(q_{\text{in}}, q_{\text{out}}))$. We define the set $\mathtt{Q}_{\div_i} = \mathtt{Q}_{-_i} \cup \mathtt{Q}_{\mathtt{T}_i}$ and the instructions $\mathtt{P}_{\div_i}(q_{\text{in}}, q_{\text{out}}) = \mathtt{P}_{T_i}(q_{\text{out}}) \cup \mathtt{P}_{-_iK} \cup \{\div_{\text{in}}, \div_{\text{loop}}, \div_{\text{out}}\}$, where:

$$\begin{array}{lcrcl} \div_{\text{in}} & : & q_{\text{in}} & \leq & s_0 \vee z_3 \\ \div_{\text{loop}} & : & s_K & \leq & s_0\mathtt{r}_3 \\ \div_{\text{out}} & : & s_0 & \leq & t_0 \vee z_i \end{array} .$$

We denote the computation relation on the divide by K program by $\leq_{\div_i(q_{\text{in}}, q_{\text{out}})}$, but will write $\leq_{\div_i}$ when understood in context. Similar to the multiply by K program, the initial instruction $\div_{\text{in}}$ is meant to verify that register $\mathtt{r}_3$ is empty and initiates the division process. That is, a block of K tokens are removed from $\mathtt{r}_i$ and 1 token is added to $\mathtt{r}_3$ repeatedly until $\mathtt{r}_i$ is empty. If $\mathtt{r}_i$ was emptied at state s_0, then $\times_3$ transfers the tokens in $\mathtt{r}_3$ to $\mathtt{r}_i$. This

can only happen if the original number of tokens in $\mathtt{r}_i$ was divisible by K, otherwise the computation would stop at some configuration labeled by a state s_δ where $0 < \delta < K$.

Proposition 5.1.5. Let $\div_i(q_{\text{in}}, q_{\text{out}})$ be a divide by K program for some $i \in \{1, 2\}$ and $0 \leq \delta \leq K$. Then $s_\delta \mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\div_i} q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$, witnessed by a computation with no instance of instruction $\div_{\text{in}}$, if and only if $m_j = n_j$, $m_3 = 0$, $K \mid (n_i+\delta)$ and $m_i = n_3+\frac{n_i+\delta}{K}$. Hence $q_{\text{in}}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\div_i} q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$, witnessed by a computation with precisely one instance of instruction $\div_{\text{in}}$, if and only if $m_j = n_j$, $m_3 = n_3 = 0$, $K \mid n_i$ and $m_i = \frac{n_i}{K}$.

Proof. Note that the only instruction that outputs the state q_{out} is $T_{\text{out}} \in \mathtt{P}_{\mathtt{T}_i}$, which is only applicable to an ID containing a configuration labeled by state t_0 in the subprogram $\mathtt{T}_i$. Now, the only instruction in $\mathtt{P}_{\div_i}$ that outputs a state in $\mathtt{Q}_{\mathtt{T}_i}$ is $\div_{\text{out}}$ which is only applicable to an ID containing a configuration labeled by state s_0. Since the only instructions applicable to a state $s_{\delta'}$, for $0 < \delta' \leq K$, are those from $\mathtt{P}_{-_iK} \cup \{\div_{\text{loop}}\}$, none of which are forking instructions, we obtain $s_\delta x \sqsubseteq_{\div_i} q_{\text{out}}x'$ iff there is a computation

$$s_\delta x \leq_{\div_i} s_0x_1 \sqsubseteq_{\div_i} t_0x_2 \sqsubseteq_{\mathtt{T}_i} q_{\text{out}}x',$$

for some $x, x', x_1, x_2 \in \mathtt{R}_3^*$.

Without loss of generality, assume $i = 1$. We proceed by induction on n_1. Suppose $n_1 = 0$. If $\delta = 0$ then the only applicable instructions are $-_1$ and $\div_{\text{out}}$. We observe then

$$s_0\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\{-_1, \div_{\text{out}}\}} qx \iff q = t_0 \,\&\, x = \mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3},$$

by Proposition 4.3.6 and since $n_1 = 0$. By Proposition 5.1.1, $t_0\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\mathtt{T}_1} q_{\text{out}}x$ iff $x = \mathtt{r}_1^{n_3}\mathtt{r}_2^{n_2}$, and the claim is satisfied. If $\delta \neq 0$, then only instructions applicable are those

from $\mathtt{P}_{-_i K} \cup \{\div_{\text{loop}}\}$. Furthermore,

$$s_\delta \mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \leq_{-_1 K} s_K x \iff \delta = K \text{ and } x = \mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \text{ by Prop. 5.1.4.}$$

Now, the only instruction applicable to $s_K x$ is $\div_{\text{loop}}$, and

$$s_K \mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \leq^{\div_{\text{loop}}} s_0 x \iff x = \mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+1}.$$

Thus, by the above,

$$s_0 \mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+1} \sqsubseteq_{\mathtt{T}_1} q_{\text{out}} x \iff x = \mathtt{r}_1^{n_3+1}\mathtt{r}_2^{n_2}.$$

Note that there are no instructions applicable to a configuration with state q_{out} except $\div_{\text{in}}$ in the case that $q_{\text{in}} = q_{\text{out}}$. Hence $s_\delta \mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\div_1} q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$ if and only if $m_2 = n_2$, $m_3 = 0$, and $m_1 = n_3 + \frac{\delta}{K}$.

Now suppose the claim holds for some $n_1 \geq 0$. For $\delta = 0$, the only applicable instructions are $-_1$ and $\div_{\text{out}}$, where we observe

$$s_0 \mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\{-_1, \div_{\text{out}}\}} qx \iff q = s_1 \,\&\, x = \mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3},$$

since $z_1 \mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \notin \text{Acc}(\emptyset)$ by Proposition 4.3.4. By the induction hypothesis, we have $s_1 \mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\div_1} q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$ iff $m_2 = n_2$, $m_3 = 0$, and $K \mid (n_1 + 1)$ with $m_1 = \frac{n_1+1}{K} + n_3$, and we are done.

For $0 < \delta < K$, the only instruction applicable is $-_{\delta+1}$ in $\mathtt{P}_{-_1 K}$, and

$$s_\delta \mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \leq^{-_{\delta+1}} s_{\delta+1}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3}.$$

By the induction hypothesis, $s_{\delta+1}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\div_i} q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$ iff $m_2 = n_2$, $m_3 = 0$, and $K \mid (n_1 + \delta + 1)$ with $m_1 = \frac{n_1+\delta+1}{K} + n_3 = \frac{(n_1+1)+\delta}{K} + n_3$, and we are done.

If $\delta = K$, then the only instruction applicable is $\div_{\text{loop}}$, and we observe

$$s_K\mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \leq^{\div_{\text{loop}}} s_0\mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+1}.$$

By the first case, $s_0\mathtt{r}_1^{n_1+1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3+1} \sqsubseteq_{\div_1} q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$ iff $m_2 = n_2$, $m_3 = 0$, and $K \mid (n_1 + 1)$ with $m_1 = \frac{n_1+1}{K} + n_3 + 1 = \frac{(n_1+1)+K}{K} + n_3$, and we are done.

Lastly, the only instruction applicable to $q_{\text{in}}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3}$ is $\div_{\text{in}}$, hence

$$q_{\text{in}}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\{\div_{\text{in}}\}} s_0\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\div} q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$$

if and only if $n_3 = m_3 = 0$, $m_2 = n_2$, $K \mid n_1$ and $m_1 = \frac{n_1}{K}$ by the above. □

Lastly, we define the *end program*, denoted by $F = (\mathtt{R}_3, \mathtt{Q}_F, \mathtt{P}_F)$ to be a transition of the final state q_f of $\mathtt{B}$ to the final state q_F of $\mathtt{B}_K$. We define $\mathtt{Q}_F = \{c_F, q_F\}$ and the instructions $\mathtt{P}_F = \{F_1, F_2\}$ are the following pair:

$$\begin{array}{rcl} F_1 & : & q_f\mathtt{r}_1 \leq c_F \\ F_2 & : & c_F\mathtt{r}_2 \leq q_F \end{array}.$$

We define $\leq_F$ to be the computation relation on the end program.

Proposition 5.1.6. $q\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \in \text{Acc}(F)$ if and only if $n_3 = 0$ and (i) $n_1 = 1$ and $n_2 = 1$ for $q = q_f$, (ii) $n_1 = 0$ and $n_2 = 1$ for $q = c_F$, and (iii) $n_1 = n_2 = 0$ for $q = q_F$.

We can now formally define the $\mathtt{B}_K$ machine. For a 2-ACM $\mathtt{B} = (\mathtt{R}_2, \mathtt{Q}, \mathtt{P})$, define $\mathtt{P}_+$, $\mathtt{P}_-$, and $\mathtt{P}_\vee$ to be the sets of increment, decrement, and forking instructions, respectively, from $\mathtt{P}$. Hence $\mathtt{P} = \mathtt{P}_+ \cup \mathtt{P}_- \cup \mathtt{P}_\vee$ is a disjoint union. Assume $\mathtt{Q}_\emptyset$ and $\mathtt{Q}_F$ are disjoint

from $\mathtt{Q}$, and for each increment and decrement instruction $p_+ : q_{\text{in}} \leq q_{\text{out}}\mathtt{r}_i \in \mathtt{P}_+$ and $p_- : q_{\text{in}}\mathtt{r}_i \leq q_{\text{out}} \in \mathtt{P}_-$, for $i \in \{1,2\}$, we relabel the elements of the following sets

$$\mathtt{Q}^{p_+}_{\times_i} := \{q^{p_+} : q \in \mathtt{Q}_{\times_i}\} \quad \& \quad \mathtt{Q}^{p_-}_{\div_i} := \{q^{p_-} : q \in \mathtt{Q}_{\div_i}\}$$
$$\mathtt{P}^{p_+}_{\times_i} := \{p^{p_+} : p \in \mathtt{P}_{\times_i}(q_{\text{in}}, q_{\text{out}})\} \quad \& \quad \mathtt{P}^{p_-}_{\div_i} := \{p^{p_-} : p \in \mathtt{P}_{\div}(q_{\text{in}}, q_{\text{out}})\},$$

making the sets disjoint.

Definition 5.1.1. Let $\mathtt{B} = (\mathtt{R}_2, \mathtt{Q}, \mathtt{P})$ be a 2-ACM and fix $K > 1$. We define the machine $\mathtt{B}_K := (\mathtt{R}_3, \mathtt{Q}_K, \mathtt{P}_K)$, where

- $\mathtt{Q}_K := \mathtt{Q} \cup \mathtt{Q}_\emptyset \cup \mathtt{Q}_F \cup \bigcup_{p \in \mathtt{P}_+} \mathtt{Q}^p_\times \cup \bigcup_{p \in \mathtt{P}_-} \mathtt{Q}^p_\div$,
- $\mathtt{P}_K := \mathtt{P}_\vee \cup \mathtt{P}_\emptyset \cup \mathtt{P}_F \cup \bigcup_{p \in \mathtt{P}_+} \mathtt{P}^p_\times \cup \bigcup_{p \in \mathtt{P}_-} \mathtt{P}^p_\div$.

Lemma 5.1.7. Let $p \in \mathtt{P}_+ \cup \mathtt{P}_-$ be an instruction acting on register $\mathtt{r}_i$, where $\{i,j\} = \{1,2\}$. Let $\mathtt{C} = q\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \in \mathtt{Conf}(\mathtt{B}_K)$ and suppose $\mathtt{C} \in \mathrm{Acc}(\mathtt{B}_K)$ witnessed by

$$\mathtt{C} \leq^{p_1} u_1 \leq^{p_2} \cdots \leq^{p_N} u_N = \mathtt{u}_F \in \mathtt{Fin}(\mathtt{B}_K).$$

1. If $p \in \mathtt{P}_+$ and $q \in \mathtt{Q}^p_\times$, then there exists $k \leq N$ such that $u_k = \mathtt{D} \vee u$ with $\mathtt{C} \sqsubseteq_\times \mathtt{D} = q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$, where $m_3 = 0$, $m_j = n_j$, and

   (i) $m_i = n_i + n_3 + \delta$, if $q = t^p_\delta$ for $\delta \in \{0,1\}$;

   (ii) $m_i = Kn_i + n_3 + K - \delta$, if $q = a^p_\delta$ for $0 \leq \delta \leq K$.

2. If $p \in \mathtt{P}_-$ and $q \in \mathtt{Q}^p_\div$, then there exists $k \leq N$ such that $u_k = \mathtt{D} \vee u$ with $\mathtt{C} \sqsubseteq_\div \mathtt{D} = q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$, where $m_3 = 0$, $m_j = n_j$, and

   (i) $m_i = n_i + n_3 + \delta$, if $q = t^p_\delta$ for $\delta \in \{0,1\}$;

   (ii) $K \mid (n_i + K - \delta)$ and $m_i = n_3 + \frac{n_i + K - \delta}{K}$, if $q = s^p_\delta$ for $0 \leq \delta \leq K$.

Proof. For (1), suppose $p \in \mathtt{P}_+$. Then there exists a multiply by K program $\times_i^p(q_{\text{in}}, q_{\text{out}})$ in $\mathtt{B}_K$. Since the only instructions applicable to $\mathtt{C}$ are those from $\mathtt{P}_\times^p$, none of which with outgoing state q_F, there must exist a smallest $k \leq N$ such that $p_k = T_3^p$ and $u_k = \mathtt{D} \vee u$ for some $u \in \mathtt{ID}(\mathtt{B}_K)$ and $\mathtt{D} = q_{\text{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}\mathtt{r}_3^{m_3}$. Since each instruction of $\mathtt{P}_\times^p \setminus \{T_{\text{out}}^p\}$ only outputs states that are in $\mathtt{Q}_\times^p \cup \mathtt{Q}_\emptyset$, the instructions $\{p_1, \ldots, p_k\} \subseteq \mathtt{P}_\times^p \cup \mathtt{P}_\emptyset$. Hence $u \in \mathtt{ID}(\emptyset)$, and since $u_k \in \mathrm{Acc}(\mathtt{B}_K)$, it follows that $u \in \mathrm{Acc}(\emptyset)$ and $\mathtt{C} \sqsubseteq_{\times \cup \emptyset} \mathtt{D}$. Since there are no $\mathtt{Q}_\emptyset$-instructions in $\mathtt{P}_\times^p$, it follows that $\mathtt{C} \sqsubseteq_\times \mathtt{D}$. Since k is minimal and $q \notin \mathtt{Q}$, it must be that $\times_{\text{in}}^p \notin \{p_1, \ldots, p_k\}$. Therefore the values of m_1, m_2, m_3 are determined by Propositions 5.1.1 and 5.1.3.

By the same argument, (2) follows with the values of m_1, m_2, m_3 and conditions on n_i determined by Propositions 5.1.1 and 5.1.5. □

Lemma 5.1.8. Let $\mathtt{C} = q_{\text{in}}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \in \mathtt{Conf}(\mathtt{B}_K)$ and suppose $q_{\text{in}} \in \mathtt{Q}$. Then $\mathtt{C} \in \mathrm{Acc}(\mathtt{B}_K)$ if and only if there exists $\mathtt{C}' \in \mathtt{Conf}(\mathtt{B})$ such that $\mathtt{C}' \in \mathrm{Acc}(\mathtt{B})$ and $\mathtt{C} = \mathtt{C}'_K$.

Proof. Let $\mathtt{C} = q_{\text{in}}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \in \mathtt{Conf}(\mathtt{B}_K)$ be given such that $q_{\text{in}} \in \mathtt{Q}$.

($\Leftarrow$) Suppose there exists $\mathtt{C}' \in \mathtt{Conf}(\mathtt{B})$ such that $\mathtt{C} = \mathtt{C}'_K$ and $\mathtt{C}' \in \mathrm{Acc}(\mathtt{B})$. Then there exists $N \in \mathbb{N}$, $\mathtt{u}_0, \ldots, \mathtt{u}_N \in \mathtt{ID}(\mathtt{B})$, and $p_1, \ldots, p_N \in \mathtt{P}$ such that

$$\mathtt{C}' = \mathtt{u}_0 \leq^{p_1} \mathtt{u}_1 \leq^{p_2} \cdots \leq^{p_N} \mathtt{u}_N = \mathtt{u}_f \in \mathtt{Fin}(\mathtt{B}).$$

We proceed by induction on N. If $N = 0$, then $\mathtt{C}' = q_f$. Hence $\mathtt{C} = \mathtt{C}'_K = q_f\mathtt{r}_1\mathtt{r}_2$. By Proposition 5.1.6(i),

$$\mathtt{C} = q_f\mathtt{r}_1\mathtt{r}_2 \in \mathrm{Acc}(F) \subset \mathrm{Acc}(\mathtt{B}_K).$$

Now let $N \geq 1$ and suppose the claim holds for all $k < N$. Let $\mathtt{C}' = q_{\text{in}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}$, then $\mathtt{C} = q_{\text{in}}\mathtt{r}_1^{Km_1}\mathtt{r}_2^{Km_2}$. We have three cases.

Case 1: Suppose p_1 is the increment instruction, without loss of generality, on register $\mathtt{r}_1$ given by $q_{\text{in}} \leq q_{\text{out}}\mathtt{r}_1$. Then $\mathtt{u}_1 = q_{\text{out}}\mathtt{r}_1^{m_1+1}\mathtt{r}_2^{m_2}$. Since $\mathtt{u}_1 \in \text{Acc}(\mathtt{B}_K)$ and has a computation of length $N-1$, by the induction hypothesis it follows that

$$q_{\text{out}}\mathtt{r}_1^{K^{m_1+1}}\mathtt{r}_2^{K^{m_2}} \in \text{Acc}(\mathtt{B}_K).$$

By the definition of $\mathtt{P}_K$, $\mathtt{P}^p_{\times_1} \subset \mathtt{P}_K$ and thus,

$$\begin{array}{lll} \mathtt{C} = q_{\text{in}}\mathtt{r}_1^{K^{m_1}}\mathtt{r}_2^{K^{m_2}} & \sqsubseteq_{\times_1^p} & q_{\text{out}}\mathtt{r}_1^{K^{m_1+1}}\mathtt{r}_2^{K^{m_2}} \quad \text{by Prop. 5.1.3} \\ & \in & \text{Acc}(\mathtt{B}_K) \quad \text{by induction hyp.} \end{array}$$

Case 2: Suppose p_1 is the decrement instruction, without loss of generality, on register $\mathtt{r}_1$ given by $q_{\text{in}}\mathtt{r}_1 \leq q_{\text{out}}$. Then $m_1 \geq 1$ and $\mathtt{u}_1 = q_{\text{out}}\mathtt{r}_1^{m_1-1}\mathtt{r}_2^{m_2}$. Since $\mathtt{u}_1 \in \text{Acc}(\mathtt{B})$ has a computation of length $N-1$, by the induction hypothesis it follows that

$$q_{\text{out}}\mathtt{r}_1^{K^{m_1-1}}\mathtt{r}_2^{K^{m_2}}\,\text{Acc}(\mathtt{B}_K).$$

By the definition of $\mathtt{P}_K$, $\mathtt{P}^p_{\div_1} \subset \mathtt{P}_K$ and thus,

$$\begin{array}{lll} \mathtt{C} = q_{\text{in}}\mathtt{r}_1^{K^{m_1}}\mathtt{r}_2^{K^{m_2}} & \sqsubseteq_{\div_1^p} & q_{\text{out}}\mathtt{r}_1^{K^{m_1-1}}\mathtt{r}_2^{K^{m_2}} \quad \text{by Prop. 5.1.5,} \\ & \in & \text{Acc}(\mathtt{B}_K) \quad \text{by induction hyp.} \end{array}$$

Case 3: Suppose p_1 is the forking instruction given by $q_{\text{in}} \leq q' \vee q''$. Then $\mathtt{u}_1 = q'\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2} \vee q''\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2}$. Hence $q'\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2} \in \text{Acc}(\mathtt{B})$ and $q''\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2} \in \text{Acc}(\mathtt{B})$. Since $\mathtt{u}_1 \in \text{Acc}(\mathtt{B})$ and has a computation of length $N-1$, so do the computations above, and by the induction hypothesis and the compatibility of $\leq_{\mathtt{B}_K}$ with $\vee$, it follows that

$$q'\mathtt{r}_1^{K^{m_1}}\mathtt{r}_2^{K^{m_2}} \vee q''\mathtt{r}_1^{K^{m_1}}\mathtt{r}_2^{K^{m_2}} \in \text{Acc}(\mathtt{B}_K).$$

Since $p_1 \in \mathtt{P}_K$ by definition, it follows that $\mathtt{C} \in \mathrm{Acc}(\mathtt{B}_K)$.

($\Rightarrow$) Suppose $\mathtt{C} \in \mathrm{Acc}(\mathtt{B}_K)$. Then there exists $N \in \mathbb{N}$, $\mathtt{u}_0, \ldots, \mathtt{u}_N \in \mathtt{ID}(\mathtt{B})$, and $p_1, \ldots, p_N \in \mathtt{P}_K$ such that

$$\mathtt{C} = \mathtt{u}_0 \leq^{p_1} \mathtt{u}_1 \leq^{p_2} \cdots \leq^{p_N} \mathtt{u}_N = \mathtt{u}_F \in \mathtt{Fin}(\mathtt{B}_K).$$

Since $q_{\text{in}} \in \mathtt{Q}$, the smallest $N \geq 2$. We proceed by induction on N. If $N = 2$, then $q_{\text{in}} = q_f$ and p_1 is the initial instruction of the end-program, which halts iff $\mathtt{C} = q_f \mathtt{r}_1 \mathtt{r}_2$ by Proposition 5.1.6. Then $\mathtt{C}' = q_f \in \mathtt{Conf}(\mathtt{B})$ is such that $\mathtt{C} = \mathtt{C}'_K$ and $\mathtt{C}' \in \mathrm{Acc}(\mathtt{B})$ by reflexivity of $\leq_{\mathtt{B}}$. So suppose $N > 2$ and the claim holds for all $k < N$. Since $q_{\text{in}} \in \mathtt{Q} \backslash \{q_f\}$, there exists an instruction $p \in \mathtt{P}$ such that either $p_1 = p \in \mathtt{P}_\vee$, $p_1 = \times^p_{\text{in}} \in \mathtt{P}^p_\times$, or $p_1 = \div^p_{\text{in}} \in \mathtt{P}^p_\div$.

Case 1: Suppose $p_1 \in \mathtt{P}_\vee$ is a forking instruction $q_{\text{in}} \leq q' \vee q''$. Since $\mathtt{P}_\vee \subset \mathtt{P}$, we obtain

$$\mathtt{u}_1 = q' \mathtt{r}_1^{n_1} \mathtt{r}_2^{n_2} \mathtt{r}_3^{n_3} \vee q'' \mathtt{r}_1^{n_1} \mathtt{r}_2^{n_2} \mathtt{r}_3^{n_3} \in \mathrm{Acc}(\mathtt{B}_K),$$

and so $q' \mathtt{r}_1^{n_1} \mathtt{r}_2^{n_2} \mathtt{r}_3^{n_3} \in \mathrm{Acc}(\mathtt{B}_K)$ and $q'' \mathtt{r}_1^{n_1} \mathtt{r}_2^{n_2} \mathtt{r}_3^{n_3} \in \mathrm{Acc}(\mathtt{B}_K)$, with computations less than N. By the induction hypothesis, $n_1 = K^{m_1}$, $n_2 = K^{m_2}$, $n_3 = 0$, where $q' \mathtt{r}_1^{m_1} \mathtt{r}_2^{m_2} \in \mathrm{Acc}(\mathtt{B})$ and $q'' \mathtt{r}_1^{m_1} \mathtt{r}_2^{m_2} \in \mathrm{Acc}(\mathtt{B})$. Thus $\mathtt{C} = \mathtt{C}'_K$ where $\mathtt{C}' = q \mathtt{r}_1^{m_1} \mathtt{r}_2^{m_2}$, and

$$\mathtt{C}' = q \mathtt{r}_1^{m_1} \mathtt{r}_2^{m_2} \leq^{p_1} q' \mathtt{r}_1^{m_1} \mathtt{r}_2^{m_2} \vee q'' \mathtt{r}_1^{m_1} \mathtt{r}_2^{m_2} \in \mathrm{Acc}(\mathtt{B}),$$

and therefore $\mathtt{C}' \in \mathrm{Acc}(\mathtt{B})$.

Case 2: Suppose $p_1 = \times^p_{\text{in}}$, where $p \in \mathtt{P}_+$ is some increment instruction $q_{\text{in}} \leq q_{\text{out}} \mathtt{r}_i$. Without loss of generality, suppose $i = 1$. Now,

$$\mathtt{C} = q_{\text{in}} \mathtt{r}_1^{n_1} \mathtt{r}_2^{n_2} \mathtt{r}_3^{n_3} \leq^{p_1} a^p_K \mathtt{r}_1^{n_1} \mathtt{r}_2^{n_2} \mathtt{r}_3^{n_3} \vee z_3 \mathtt{r}_1^{n_1} \mathtt{r}_2^{n_2} \mathtt{r}_3^{n_3} \in \mathrm{Acc}(\mathtt{B}_K)$$

so $n_3 = 0$ by Proposition 4.3.4 and $a_K^p \mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \in \mathrm{Acc}(\mathtt{B}_K)$ by Proposition 4.1.12. Hence, by Lemma 5.1.7(1), there exists $1 < k < N$ such that $\mathtt{u}_k = \mathtt{D} \vee \mathtt{u}$ and $\mathtt{C} \sqsubseteq_{\times_1^p} \mathtt{D} = q_{\mathrm{out}}\mathtt{r}_1^{Kn_1}\mathtt{r}_2^{n_2}$. Since $\mathtt{u}_k \in \mathrm{Acc}(\mathtt{B}_K)$ has a computation of length less than N, by the induction hypothesis it follows that there is $\mathtt{D}' \in \mathtt{Conf}(\mathtt{B})$ such that $\mathtt{D} = \mathtt{D}'_K$, i.e. $Kn_1 = K^{m_1+1}$ and $n_2 = K^{m_2}$, and $\mathtt{D}' = q_{\mathrm{out}}\mathtt{r}_1^{m_1+1}\mathtt{r}_2^{m_2} \in \mathrm{Acc}(\mathtt{B})$. Let $\mathtt{C}' = q_{\mathrm{in}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2} \in \mathtt{Conf}(\mathtt{B})$. Therefore $\mathtt{C} = \mathtt{C}'_K$ and, since $\mathtt{C}' \leq^p \mathtt{D}'$, $\mathtt{C}' \in \mathrm{Acc}(\mathtt{B})$.

Case 3: Suppose $p_1 = \div_{\mathrm{in}}^p$, where $p \in \mathtt{P}_-$ some decrement instruction $q_{\mathrm{in}}\mathtt{r}_i \leq q_{\mathrm{out}}$. Without loss of generality, suppose $i = 1$. Now,

$$\mathtt{C} = q_{\mathrm{in}}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \leq^{p_1} s_0^p\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \vee z_3\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \in \mathrm{Acc}(\mathtt{B}_K),$$

so $n_3 = 0$ by Proposition 4.3.4 and $s_0^p\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \in \mathrm{Acc}(\mathtt{B}_K)$ by Proposition 4.1.12. Hence, by Lemma 5.1.7(2), there exists $1 < k < N$ such that $\mathtt{u}_k = \mathtt{D} \vee \mathtt{u}$, $K \mid n_1$, and $\mathtt{C} \sqsubseteq_{\div_1^p} \mathtt{D} = q_{\mathrm{out}}\mathtt{r}_1^{n_1/K}\mathtt{r}_2^{n_2}$. Since $\mathtt{u}_k \in \mathrm{Acc}(\mathtt{B}_K)$ has a computation of length less than N, by the induction hypothesis it follows that there is $\mathtt{D}' \in \mathtt{Conf}(\mathtt{B})$ such that $\mathtt{D} = \mathtt{D}'_K$, i.e. $\frac{n_1}{K} = K^{m_1}$ and $n_2 = K^{n_2}$, and $\mathtt{D}' = q_{\mathrm{out}}\mathtt{r}_1^{m_1}\mathtt{r}_2^{m_2} \in \mathrm{Acc}(\mathtt{B})$. Let $\mathtt{C}' = q_{\mathrm{in}}\mathtt{r}_1^{m_1+1}\mathtt{r}_2^{m_2} \in \mathtt{Conf}(\mathtt{B})$. Therefore $\mathtt{C} = \mathtt{C}'_K$ and, since $\mathtt{C}' \leq^p \mathtt{D}'$, we obtain $\mathtt{C}' \in \mathrm{Acc}(\mathtt{B})$. □

Let $\tilde{\mathtt{B}}$ be the 2-ACM given by Theorem 4.3.8. Since membership of $\mathrm{Acc}(\tilde{\mathtt{B}})$ is undecidable, we obtain the following:

Corollary 5.1.9. Membership of the set $\mathrm{Acc}(\tilde{\mathtt{B}}_K)$ is undecidable for $K > 1$.

5.1.3 Simple equations and admissibility for ACMs.

When writing simple equations, we will be using the set of variables $\{x_i\}_{i\in\mathbb{Z}^+}$, and we will assume implicitly that this set is ordered by the natural order of the indices. We also define $\mathbf{x_n} := (x_1, \ldots, x_n)$, for all $n \in \mathbb{Z}^+$ and for a tuple $\overline{a} = (a_1, \ldots, a_n)$ of natural numbers, we define $\mathbf{x_n}^{\overline{a}} = x_1^{a_1} \cdots x_n^{a_n}$; we also define $\mathbf{x_n}^{\mathbf{1}} = x_1 \cdots x_n$. In this way, any commutative monoid term is of the form $\mathbf{x_n}^{\overline{a}}$, and thus it is fully specified by such an $\overline{a}$.

Let $[\mathrm{R}] = (\mathbf{1_n}, \mathrm{R})$ be a simple equation. Since, for any k-ACM $\mathtt{B}$, $\mathbf{W}^+_{\mathtt{B}}$ is commutative (Lemma 4.3.2), $\mathbf{W}^+_{\mathtt{B}} \models [\mathrm{R}]$ iff $\mathbf{W}^+_{\mathtt{B}} \models [\mathrm{R}]_{\mathrm{com}}$, where

$$[\mathrm{R}]_{\mathrm{com}} : \mathbf{x_n}^{\mathbf{1}} \leq \bigvee_{r \in \mathrm{R}} \mathbf{x_n}^{\overline{r}},$$

where $\overline{r} \in \mathbb{N}^n$ for each $r \in \mathrm{R}$ such that $\overline{r}(i) := \#(r, x_i)$. That is $\mathsf{CRL} \models [\mathrm{R}] \equiv [\mathrm{R}]_{\mathrm{com}}$. We call equations of the form $[\mathrm{R}]_{\mathrm{com}}$ the *canonical simple equations of* CRL. It will be useful to identify the set $\mathrm{R} \cup \{\mathbf{1_n}\}$, of some canonical simple equation $[\mathrm{R}] = (\mathbf{1_n}, \mathrm{R})$ of CRL, directly as a set of n-tuples with entries in $\mathbb{N}$, where $\mathbf{1_n} = (1, \ldots, 1) \in \mathbb{N}^n$.

In the following we will work interchangeably in the free monoid over the variable set $\{x_1, \ldots, x_n\}$ and also in the isomorphic monoid $\mathbb{N}^n$, for some fixed $n \geq 1$. For reasons that will be clear soon, we view the elements of $\mathbb{N}^n$ as column vectors and we also consider the bijective set $(\mathbb{N}^T)^T$ of the row vectors, which are the transposes of the elements of $\mathbb{N}^n$. In particular, for $r \in \mathbb{N}^n$ and $\sigma \in (\mathbb{N}^n)^T$, the matrix product σr yields a 1×1 matrix, which we identify with the natural number equal to its unique entry. For a set $A \subseteq \mathbb{N}^n$, we write $\sigma A := \{\sigma a \in \mathbb{N}^k : a \in A\}$.

Lemma 5.1.10. Let $[\mathrm{R}] = (\mathbf{1_n}, \mathrm{R})$ be a non-integral simple equation. Then for all $\sigma \in (\mathbb{N}^n)^T$, if $\sigma \mathrm{R} = \{0\}$ implies $\sigma = \mathbf{0}$.

Proof. Suppose $\sigma \in (\mathbb{N}^n)^T$ is such that $\sigma \neq \mathbf{0}$ but $\sigma \mathrm{R} := \{\sigma r \in \mathbb{N} : r \in \mathrm{R}\} = \{0\}$. Since $\sigma \neq 0$, there exists $i \leq n$ such that $\sigma(i) > 0$. Since $\sigma r = 0$, it must be that $r(i) = 0$ for all $r \in \mathrm{R}$. Since $\mathbf{1_n}(i) = 1$, this implies $\mathrm{supp}(\mathbf{1_n}) \setminus \mathrm{supp}(\mathrm{R}) \neq \emptyset$. Hence $[\mathrm{R}]$ is integral by Definition 2.1.1. □

Let $[\mathrm{D}] = (\mathbf{1_n}, \mathrm{D})$ be a simple equation and $K > 1$. We write $[\mathrm{D}], K \models (\star\star)$ if the following technical condition is satisfied:

$$\begin{gathered}\text{For all } \sigma, \sigma' \in (\mathbb{N}^n)^T \text{ and for all } C, C' \in \mathbb{N},\\ \text{if } C + \sigma d \text{ and } C' + \sigma' d \text{ are powers of } K \text{ for each } d \in \mathrm{D},\\ \text{then there exists } \bar{d} \in \mathrm{D} \text{ such that } \sigma\bar{d} = \sigma\mathbf{1_n} \text{ and } \sigma'\bar{d} = \sigma'\mathbf{1_n}.\end{gathered} \tag{$\star\star$}$$

For a set of simple equations Σ, by $\Sigma, K \models (\star\star)$ we mean $[\mathrm{R}], K \models (\star\star)$ for all $[\mathrm{R}] \in \Sigma$.

Lemma 5.1.11. Let Σ be a non-mingly set of simple equations, $\mathtt{B}$ be a 2-ACM, and $K > 1$. If $\Sigma, K \models (\star\star)$ then Σ is strongly admissible for $\mathtt{B}_K$.

Proof. Let $[\mathrm{D}] = (\mathbf{1_n}, \mathrm{D}) \in \Sigma$. By Lemma 4.1.11, we need only show that if $t \leq^{\mathrm{D}} \bigvee_{d\in\mathrm{D}} t_d$, then

$$\forall d \in \mathrm{D}, t_d \in \mathrm{Acc}(\mathtt{B}_K) \implies t \in \mathrm{Acc}(\mathtt{B}_K),$$

where $t, t_d \in (\mathtt{Q}_K \cup \mathtt{R}_3)^*$ for each $d \in \mathrm{D}$. So suppose $t_d \in \mathrm{Acc}(\mathtt{B}_K)$ for all $d \in \mathrm{D}$. Since $\mathrm{Acc}(M_K) \subseteq \mathtt{Conf}(\mathtt{B}_K)$, it follows that $t_d = \mathtt{C}_d \in \mathtt{Conf}(\mathtt{B}_K)$ for each $d \in \mathrm{D}$. Since Σ is not mingly, $\mathtt{B}_K$ is $\mathtt{Q}_K$-admissible by Lemma 4.1.14, which implies $t = \mathtt{C} \in \mathtt{Conf}(\mathtt{B}_K)$. We need only show $\mathtt{C} \in \mathrm{Acc}(\mathtt{B}_K)$. By definition of $\leq^{\mathrm{D}}$,

$$\mathtt{C} = qx \cdot \mathbf{x_n}^{\mathbf{1_n}} \leq^{\mathrm{D}} \bigvee_{d\in\mathrm{D}} qx \cdot \mathbf{x_n}^d = \bigvee_{d\in\mathrm{D}} \mathtt{C}_d,$$

for some $q \in \mathtt{Q}_K$ and $x, x_1, \ldots, x_n \in \mathtt{R}_3^*$, where $\mathbf{x_n} = (x_1, \ldots, x_n)$.

If $[\mathrm{D}]$ is trivial then there is $\bar{d} \in \mathrm{D}$ such that $\bar{d} = \mathbf{1_n}$, so $\mathtt{C} = \mathtt{C}_{\bar{d}} \in \mathrm{Acc}\mathtt{B}_K$ and we are done. So assume $[\mathrm{D}]$ is not trivial. Write $x = \mathtt{r}_1^{C_1}\mathtt{r}_2^{C_2}\mathtt{r}_3^{C_3}$, where $C_1, C_2, C_3 \geq 0$, and for each $j \in \{1, 2, 3\}$, define $\sigma_j \in (\mathbb{N}^n)^T$ via $\sigma_j(i) = n_j$ where $x_i = \mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3}$, for each

$i = 1, \ldots, n$. So $x_i = \mathtt{r}_1^{\sigma_1(i)}\mathtt{r}_2^{\sigma_2(i)}\mathtt{r}_3^{\sigma_3(i)}$, for each $i = 1, \ldots, n$. Thus,

$$\mathtt{C} = qx\prod_{i=1}^{n} x_i = q\mathtt{r}_1^{C_1+\sigma_1\mathbf{1_n}}\mathtt{r}_2^{C_2+\sigma_2\mathbf{1_n}}\mathtt{r}_3^{C_3+\sigma_3\mathbf{1_n}},$$

and for each $d \in \mathrm{D}$,

$$\mathtt{C}_d = qx\prod_{i=1}^{n} x_i^{d(i)} = q\mathtt{r}_1^{C_1+\sigma_1 d}\mathtt{r}_2^{C_2+\sigma_2 d}\mathtt{r}_3^{C_3+\sigma_3 d}.$$

We proceed by case analysis for each state $q \in \mathtt{Q}_K$. Suppose $q = q_F$. Since there is no q_F-instruction in $\mathtt{P}_K$, it follows that $\mathtt{C}_d = q_F$ for each $d \in \mathrm{D}$. Hence $x = 1$ and $\sigma_j\mathrm{D} = \{0\}$ for each $j \in \{1, 2, 3\}$. By Lemma 5.1.10, this implies that $\sigma_j = \mathbf{0}$ for each $j \in \{1, 2, 3\}$. Therefore $\mathtt{C} = q_F \in \mathrm{Acc}(\mathtt{B}_K)$.

Suppose $q = c_F$. By Proposition 5.1.6, $\mathtt{C}_d \in \mathrm{Acc}(F)$ iff $\mathtt{C}_d = c_F\mathtt{r}_2$. Hence $C_1 = C_3 = 0$, $\sigma_1 d = \sigma_3 d = 0$ for each $d \in \mathrm{D}$, and $C_2 + \sigma_2 d = 1$. By Lemma 5.1.10, $\sigma_1 = \sigma_3 = \mathbf{0}$, and by ($\star\star$), there exists $\bar{d} \in \mathrm{D}$ such that $C_2 + \sigma_2\bar{d} = C_2 + \sigma_2\mathbf{1_n}$. Hence $\mathtt{C} = \mathtt{C}_{\bar{d}} \in \mathrm{Acc}(\mathtt{B}_K)$.

Suppose $q = z_i$, and without loss of generality, let $i = 3$. Then for all $d \in \mathrm{D}$, $\mathtt{C}_d \in \mathrm{Acc}(\emptyset)$ iff $C_3 + \sigma_3 d = 0$ by Proposition 4.3.4. This implies $C_3 = 0$ and $\sigma_3\mathrm{D} = \{0\}$. So by Lemma 5.1.10, $\sigma_3 = \mathbf{0}$. Hence $C_3 + \sigma_3\mathbf{1_n} = 0$ and $\mathtt{C} \in \mathrm{Acc}(\emptyset) \subset \mathrm{Acc}(\mathtt{B}_K)$.

Suppose $q \in \mathtt{Q}$. By Lemma 5.1.8, for each $d \in \mathrm{D}$, $\mathtt{C}_d \in \mathrm{Acc}(\mathtt{B}_K)$ iff there exists $\mathtt{C}'_d \in \mathtt{Conf}(\mathtt{B})$ such that $\mathtt{C}_d = (\mathtt{C}'_d)_K$ with $\mathtt{C}'_d \in \mathrm{Acc}(\mathtt{B})$. I.e., for each $d \in \mathrm{D}$, $C_1 + \sigma_1 d$ and $C_2 + \sigma_2 d$ are powers of K and $C_3 + \sigma_3 d = 0$. By Lemma 5.1.10, $C_3 = 0$ and $\sigma_3 = \mathbf{0}$, and by ($\star\star$), there exists $\bar{d} \in \mathrm{D}$ such that $\sigma_1\bar{d} = \sigma_1\mathbf{1_n}$ and $\sigma_2\bar{d} = \sigma_2\mathbf{1_n}$. Therefore,

$$\mathtt{C} = q\mathtt{r}_1^{C_1+\sigma_1\mathbf{1_n}}\mathtt{r}_2^{C_2+\sigma_2\mathbf{1_n}} = q\mathtt{r}_1^{C_1+\sigma_1\bar{d}}\mathtt{r}_2^{C_2+\sigma_2\bar{d}} = \mathtt{C}_{\bar{d}} \in \mathrm{Acc}(\mathtt{B}_K).$$

Lastly, suppose $q \in \mathtt{Q}_K \setminus (\mathtt{Q} \cup \mathtt{Q}_F \cup \mathtt{Q}_\emptyset)$. Then q is an internal state of a multiply or divide by K program given by some instruction $p \in \mathtt{P}_+ \cup \mathtt{P}_-$. We can assume, without loss of generality, that p acts on register $\mathtt{r}_1$ with input state $q_{\text{in}} \in \mathtt{Q}$ and output state $q_{\text{out}} \in \mathtt{Q}$.

First, observe that

1. $\mathtt{C} \sqsubseteq_{\mathtt{T}_1^p} \mathtt{D} = q_{\text{out}}\mathtt{r}_1^{(C_1+\sigma_1\mathbf{1_n})+(C_3+\sigma_3\mathbf{1_n})+\delta}\mathtt{r}_2^{C_2+\sigma_2\mathbf{1_n}}$ if $q = t_\delta^p$ by Proposition 5.1.1,
2. $\mathtt{C} \sqsubseteq_{\mathtt{\times}_1^p} \mathtt{D} = q_{\text{out}}\mathtt{r}_1^{K(C_1+\sigma_1\mathbf{1_n}+K-\delta)+(C_3+\sigma_3)\mathbf{1_n}}\mathtt{r}_2^{C_2+\sigma_2\mathbf{1_n}}$ if $q = a_\delta^p$ by Proposition 5.1.3,
3. $\mathtt{C} \sqsubseteq_{\mathtt{\div}_1^p} \mathtt{D} = q_{\text{out}}\mathtt{r}_1^{(C_3+\sigma_3\mathbf{1_n})+(C_1+\sigma_1\mathbf{1_n}+K-\delta)/K}\mathtt{r}_2^{C_2+\sigma_2\mathbf{1_n}}$ if $q = s_\delta^p$ and $K \mid (C_1 + \sigma_1\mathbf{1_n} + K - \delta)$ by Proposition 5.1.5.

Now, for each $d \in \mathrm{D}$ and $j \in \{1, 2, 3\}$, we define $n_j^d := C_j + \sigma_j d$. Since $\mathtt{C}_d \in \mathrm{Acc}(\mathtt{B}_K)$ for all $d \in \mathrm{D}$, by Lemma 5.1.7 it follows that $\mathtt{C}_d \sqsubseteq_{\mathtt{B}_K} \mathtt{D}_d$ and $\mathtt{D}_d \in \mathrm{Acc}(\mathtt{B}_K)$, where $\mathtt{D}_d := q_{\text{out}}\mathtt{r}_1^{m_1^d}\mathtt{r}_2^{m_2^d}$ with $m_2^d = n_2^d$ and

1. $m_1^d = n_1^d + n_3^d + \delta$ if $q = t_\delta^p$ for $\delta \in \{0, 1\}$,
2. $m_1^d = Kn_1^d + n_3^d + K - \delta$ if $q = a_\delta^p$ for $0 \le \delta \le K$, and
3. $K \mid (n_1^d + K - \delta)$ and $m_1^d = n_3^d + (n_1^d + K - \delta)/K$ if $q = s_\delta^p$ for $0 \le \delta \le K$.

Furthermore, since $q_{\text{out}} \in \mathtt{Q}$, by Lemma 5.1.8, m_1^d and m_2^d are powers of K for each $d \in \mathrm{D}$ (and thus Km_1^d is as well). Note that $K \cdot (\sigma d) = (K \cdot \sigma)d$, for any $\sigma \in (\mathbb{N}^n)^T$. So for each $d \in \mathrm{D}$ we observe,

$$\begin{aligned} n_1^d + n_3^d + \delta &= (C_1 + C_3 + \delta) + (\sigma_1 + \sigma_3)d \\ Kn_1^d + n_3^d + K - \delta &= (KC_1 + C_3 + K - \delta) + (K\sigma_1 + \sigma_3)d \ , \\ n_1^d + Kn_3^d + K - \delta &= (C_1 + KC_3 + K - \delta) + (\sigma_1 + K\sigma_3)d \end{aligned}$$

for any $\delta \leq K$. Since m_1^d and m_2^d are powers of K (and thus Km_1^d), for each $d \in \mathrm{D}$, by ($\star\star$) there exists $\bar{d} \in \mathrm{D}$ such that $\sigma_2\bar{d} = \sigma_2\mathbf{1_n}$ and $\sigma\bar{d} = \sigma\mathbf{1_n}$, where

$$\sigma \in \{\sigma_1 + \sigma_3, K\sigma_1 + \sigma_3, \sigma_1 + K\sigma_3\}.$$

It immediately follows that $\mathtt{C} \sqsubseteq_{\mathtt{T}_1^p} \mathtt{D} = \mathtt{D}_{\bar{d}}$ if $q = t_\delta^p$, and $\mathtt{C} \leq_{\times_1^p} \mathtt{D} = \mathtt{D}_{\bar{d}}$ if $q = a_\delta^p$. For $q = s_\delta^p$, we need only show that $K \mid (C_1 + \sigma_1\mathbf{1_n} + K - \delta)$. Now, $m_1^{\bar{d}} = K^t$ for some $t \geq 0$, hence

$$\begin{aligned} Km_1^{\bar{d}} = K^{t+1} &= (C_1 + KC_3 + K - \delta) + (\sigma_1 + K\sigma_3)\bar{d} \\ &= (C_1 + KC_3 + K - \delta) + (\sigma_1 + K\sigma_3)\mathbf{1_n} \\ &= (C_1 + \sigma_1\mathbf{1_n} + K - \delta) + K(\sigma_3\mathbf{1_n}) + KC_3 \\ \implies K^t &= \tfrac{1}{K}(C_1 + \sigma_1\mathbf{1_n} + K - \delta) + \sigma_3\mathbf{1_n} + C_3, \end{aligned}$$

and since K^t, C_3, and $\sigma_3\mathbf{1_n}$ are integers, it follows that $K \mid (C_1 + \sigma_1\mathbf{1_n} + K - \delta)$. Therefore, by Proposition 5.1.5, $\mathtt{C} \sqsubseteq_{\div_1^p} \mathtt{D} = \mathtt{D}_{\bar{d}}$. In any case $\mathtt{C} \sqsubseteq_{\mathtt{B}_K} \mathtt{D}_{\bar{d}} \in \mathrm{Acc}(\mathtt{B}_K)$, and therefore $\mathtt{C} \in \mathrm{Acc}(\mathtt{B}_K)$. □

5.1.4 Undecidability, the class $\mathcal{U}$, and spinal equations.

An ISR-equation $[V] = (f, V)$ is called *spinal* if $[V]$ is of the form:

$$[V] : \underbrace{x_1^{f(1)} \cdots x_k^{f(k)}}_{f} \leq \underbrace{(1 \vee)}_{\mathbf{0}} \underbrace{x_1^{v_1(1)}}_{v_1} \vee \underbrace{x_1^{v_2(1)} x_2^{v_2(2)}}_{v_2} \vee \cdots \vee \underbrace{x_1^{v_k(1)} \cdots x_k^{v_k(k)}}_{v_k},$$

where $f \notin V$ and $(1\vee)$ is meant to signify 1 may or may not be included in the join. In this way, a simple equation $[\mathrm{R}]$ is *pre-spinal* if there exists a substitution $\boldsymbol{\sigma}$ such that $[\boldsymbol{\sigma}\mathrm{R}]$ is equivalent, modulo commutativity, to a spinal equation $[V]$ (written $[V] = [\boldsymbol{\sigma}\mathrm{R}]_{\mathrm{com}}$).

Example 5.1.1. Let $[\mathrm{R}], [\mathrm{D}]$ the simple equation $[\mathrm{R}] : x \leq x^2 \vee 1$ and $[\mathrm{D}] : x \leq x^2 \vee x^4$. By our definition, $[\mathrm{R}]$ is a spinal equation, while there is no substitution on $[\mathrm{D}]$ that can result in a spinal equation, so $[\mathrm{D}]$ is not pre-spinal.

Consider the machine $\mathtt{B}_{\mathrm{even}}$. As before, it is easy to see that $q_0\mathtt{r}_1^3 \in \mathrm{Acc}(\mathrm{R}\mathtt{B}_{\mathrm{even}}) \setminus \mathrm{Acc}(\mathtt{B}_{\mathrm{even}})$. However, unlike for the equation [D] (see Section 5.1.1), this behavior cannot be controlled with $\mathtt{B}_K$ for any $K > 1$. E.g., $n = (K^4 - K^2)/2$, then $q_0\mathtt{r}_1^{K^2+n} \notin \mathrm{Acc}(\mathtt{B}_K)$ since $K^2 + n \neq K^{2m}$ for any $m \in \mathbb{N}$, however

$$q_0\mathtt{r}_1^{K^2+n} = q_0\mathtt{r}_1^{K^2}\mathtt{r}_1^n \leq^{\mathrm{R}} q_0\mathtt{r}_1^{K^2}\mathtt{r}_1^{2n} \vee q_0\mathtt{r}_1^{K^2}\mathtt{r}_1^0 = q_0\mathtt{r}_1^{K^4} \vee q_0\mathtt{r}_1^{K^2} \in \mathrm{Acc}(\mathtt{B}_K).$$

In fact, we will show this failure occurs, not just for functions of the form $n \mapsto K^n$ but actually for any (computable) injective function on $\mathbb{N}$. This is due to the fact that $[\mathrm{D}], K \models (\star\star)$ (see Theorem 5.3.1) for all sufficiently large K, but $[\mathrm{R}], K \not\models (\star\star)$ for any possible $K > 1$. To see this, note that $[\mathrm{R}] = (\mathbf{1_1}, \{r_0, r_2\})$, viewing $\mathrm{R} \subseteq \mathbb{N}^1$ where, $\mathbf{1_1} = 1$, $r_0 = 0$, and $r_2 = 2$. Then $\sigma := n \in \mathbb{N}^T$ and $C = K^2$ are such that $C + \sigma r_0 = C = K^2$ and $C + \sigma r_2 = K^2 + 2n = K^4$, both powers of K, but $\sigma\mathbf{1_1} = K^2 + n \notin \{K^2, K^4\}$, witnessing $[\mathrm{R}], K \not\models (\star\star)$.

Definition 5.1.2. Define $\mathcal{U}$ to be the class of simple equations defined via $[\mathrm{D}] \in \mathcal{U}$ if and only if $[\mathrm{D}]_{\mathrm{com}}$ is not pre-spinal.

Note that all knotted equations $[\mathsf{k}_n^m] : x^n \leq x^m$ are spinal and so their equivalent simple equations $[\mathsf{K}_n^m]$ (as defined in Section 2.4) are pre-spinal. As a consequence of the definition, $[\mathrm{R}]$ is spinal if an only if $[\mathrm{R} \cup \{\mathbf{0}\}]$ is spinal, so all equations $x^n \leq x^m \vee 1$ are spinal as well.[4] On the other hand, equations of the form $[A] : x^n \leq \bigvee_{p \in P} x^p$ is not spinal for any $n \geq 1$ and finite set $P \subseteq \mathbb{N}$ such that $|P \setminus \{0\}| \geq 2$, and it is easy to prove that the equivalent simple equation for $[A]$ is not pre-spinal.

[4]It should be noted that knotted extensions of CRL have the FEP (see Proposition 3.1.1) and hence a decidable word problem, but decidability results for $x^n \leq x^m \vee 1$ are unknown to this author.

In Section 5.3, we will show that $[\mathrm{D}], K \models (\star\star)$ for some $K > 1$ if and only if $[\mathrm{D}]$ is not *pre-spinal*. In fact, Theorem 5.3.13 states that $[\mathrm{D}]$ is not pre-spinal if and only if there exists $N \in \mathbb{N}$ such that $[\mathrm{D}], K \models (\star\star)$ for any $K \geq N$.

Therefore, by Theorem 4.3.8, Lemma 5.1.11, and Theorem 5.3.13 proved in Section 5.3, we obtain

Theorem 5.1.12. Let $\Gamma \subseteq \mathcal{U}$ be finite. Then any variety $\mathcal{V}$ in the interval $\mathsf{CRL} + \Gamma \subseteq \mathcal{V} \subseteq \mathsf{RL}$ has an undecidable word problem, particularly for its $\{\vee, \cdot, 1\}$-fragment.

As a consequence of Corollary 2.5.2, it follows that:

Theorem 5.1.13. Then $\mathsf{CRL} + \Gamma$ has an undecidable equational theory for anyfiniteand expansive $\Gamma \subseteq \mathcal{U}$.

Hence $\mathsf{CRL} + [E]$ has an undecidable equational theory, for any expansive equation

$$[E] : x^n \leq \bigvee_{p \in P} x^{n+p},$$

where $n \geq 1$ and $|P| \geq 2$. E.g., $\mathsf{CRL} + (x \leq x^2 \vee x^3)$ has an undecidable equational theory.

5.2 Admissibility for CMs

As in the previous section, we wish extend undecidability results for the $\{\leq, \cdot, 1\}$-fragment of RL to those varieties defined by simple equations. We begin as before by defining the $\mathtt{M}_K$ for an arbitrary 2-CM $\mathtt{M}$.

5.2.1 The $\mathtt{M}_K$ Machine. Proceeding as in the previous section, we will define the corresponding 3-CM $\mathtt{M}_K$ from a given 2-CM $\mathtt{M}$. We will provide sufficient conditions for when a simple equation $[\mathrm{D}]$ is admissible for $\mathtt{M}_K$, i.e., conditions which ensure that $\mathtt{C} \in \mathrm{Acc}(\mathrm{D}\mathtt{M}_K)$ iff $\mathtt{C} \in \mathrm{Acc}(\mathtt{M}_K)$, for all $\mathtt{C} \in \mathtt{Conf}(\mathtt{M}_K)$. Such equations will be closely related to the set $\mathcal{U}$ of simple equations, insofar as the encoding breaks down for pre-spinal equations as well as equations satisfying some corresponding technical weakening of commutativity.

As before, if $\mathtt{M} = (\mathtt{R}_2, \mathtt{Q}, \mathtt{P})$ is a 2-CM, then the set of instructions of $\mathtt{P}_K$ for $\mathtt{M}_K$ are obtained by replacing each increment and decrement instruction in $\mathtt{P}$ by the programs *multiply by* K and *divide by* K, respectively. However, we will need to replace zero-test instructions by K^0*-test* programs, whose implementation is meant to test whether a given register contains exactly one token (i.e., K^0) or not. That is, if $p : q_{\mathrm{in}}\mathtt{S}_i\mathtt{S}_{i+1} \leq q_{\mathrm{out}}\mathtt{S}_i\mathtt{S}_{i+1}$ is a zero-test the $\mathtt{r}_i$-register instruction and $\mathcal{P}^p$ is some program meant to simulate p in $\mathtt{M}_K$, i.e., say $i = 1$, then if $\mathtt{C} = \langle q_{\mathrm{in}}; n_1, n_2\rangle \in \mathtt{Conf}(\mathtt{M})$ and $\mathtt{C}' = \langle q_{\mathrm{out}}; m_1, m_2\rangle \in \mathtt{Conf}(\mathtt{M})$,

$$\mathtt{C} \leq^p \mathtt{C}' \iff \mathtt{C}_K = \langle q_{\mathrm{in}}; K^{n_1}, K^{n_2}, 0\rangle \leq_{\mathcal{P}^p} \langle q_{\mathrm{out}}; K^{m_1}, K^{m_2}, 0\rangle = \mathtt{C}'_K,$$

then is must be that $n_1 = m_1 = 0$ and $n_2 = m_2$, so the $\mathtt{r}_1$-register of $\mathtt{C}_K$ therefore contains precisely one $\mathtt{r}_i$-token. Using the language of counter machines, this can be achieved by

defining auxiliary states z_1^p, z_2^p and proper instructions

$$\begin{aligned} q_{\text{in}}\mathtt{S}_1\mathtt{r}_i &\leq z_1^p\mathtt{S}_1 \\ z_1^p\mathtt{S}_1\mathtt{S}_2\mathtt{r}_i &\leq z_2^p\mathtt{S}_1\mathtt{S}_2 \\ z_2^p\mathtt{S}_1 &\leq q_{\text{out}}\mathtt{S}_1\mathtt{r}_1. \end{aligned}$$

However, since we are only checking for the appearance of a specific word in a configuration, namely $\mathtt{S}_1\mathtt{r}_1\mathtt{S}_2$ for $i = 1$, we will opt to instead simulate p by a single, non-proper, instruction of the form:

$$1^p : q_{\text{in}}\mathtt{S}_i\mathtt{r}_i\mathtt{S}_{i+1} \leq q_{\text{out}}\mathtt{S}_i\mathtt{r}_i\mathtt{S}_{i+1} \tag{5.1}$$

The construction and implementation of the multiply and divide programs are essentially the same as in Section 5.1.2, with the added benefit that the zero-test program can be replaced by zero-test instructions native to the structure of counter machines. As before, these programs are defined from simpler programs named *transfer*, *add-*K, and *subtract-*K. Their intended interpretation is the exactly the same as their counterparts in Section 5.1.2. We will assume that all state names defined by the following machines are disjoint from each other, disjoint $\mathtt{Q}_\emptyset$, and disjoint from the states $\mathtt{Q}$ from a fixed 2-CM $\mathtt{M}$.

A *transfer program* $\mathtt{T}_i(q_{\text{out}}) = (\mathtt{R}_3, \mathtt{Q}_{\mathtt{T}_i}, \mathtt{P}_{\mathtt{T}_i}(q_{\text{out}}))$ is meant to transfer all contents in register $\mathtt{r}_3$ to register $\mathtt{r}_i$ and output state q_{out}. We define the set $\mathtt{Q}_{\mathtt{T}_i} = \{t_0, t_1\}$ and the set of instructions $\mathtt{P}_{\mathtt{T}_i}(q_{\text{out}}) = \{T_-, T_+, T_{\text{out}}\}$, where:

$$\begin{aligned} T_- &: t_0\mathtt{S}_3\mathtt{r}_3 \leq t_1\mathtt{S}_3 \\ T_+ &: t_1\mathtt{S}_i \leq t_0\mathtt{S}_i\mathtt{r}_i \\ T_{\text{out}} &: t_0\mathtt{S}_3\mathtt{S}_4 \leq q_{\text{out}}\mathtt{S}_3\mathtt{S}_4 \end{aligned} \quad .$$

We define $\leq_{\mathtt{T}_i}$ to be the computation relation of the transfer program $\mathtt{T}_i(q_{\text{out}})$.

The *add-K program* is denoted by $+K = (\mathtt{R}_3, \mathtt{Q}_{+K}, \mathtt{P}_{+K})$, and is intended to add K tokens to register $\mathtt{r}_3$ and output with state a_K. We define the set $\mathtt{Q}_{+K} = \{a_0, \ldots, a_K\}$, and the set of instructions $\mathtt{P}_{+K} = \{+_1, \ldots, +_K\}$, where:

$$\begin{array}{lcrcl} +_1 & : & a_0\mathtt{S}_3 & \leq & a_1\mathtt{S}_3\mathtt{r}_3 \\ +_2 & : & a_1\mathtt{S}_3 & \leq & a_2\mathtt{S}_3\mathtt{r}_3 \\ \vdots & & & \vdots & \\ +_K & : & a_{K-1}\mathtt{S}_3 & \leq & a_K\mathtt{S}_3\mathtt{r}_3 \end{array} .$$

We define $\leq_+$ to be the computation relation on the add-K program.

We now define the *multiply by K* programs, denoted by $\times_i(q_{\text{in}}, q_{\text{out}}) = (\mathtt{R}_3, \mathtt{Q}_{\times_i}, \mathtt{P}_{\times_i})$, where $i \in \{1, 2\}$. This program is meant to multiply the contents of $\mathtt{r}_i$ by K, with input state q_{in} and output state q_{out}. We define the set $\mathtt{Q}_\times = \mathtt{Q}_{+K} \cup \mathtt{Q}_{\mathtt{T}_i}$ and the set $\mathtt{P}_{\times_i} = \mathtt{P}_{\mathtt{T}_i}(q_{\text{out}}) \cup \mathtt{P}_{+K} \cup \{\times_{\text{in}}, \times_{\text{loop}}, \times_{\text{out}}\}$, where:

$$\begin{array}{lcrcl} \times_{\text{in}} & : & q_{\text{in}}\mathtt{S}_3\mathtt{S}_4 & \leq & a_K\mathtt{S}_3\mathtt{S}_4 \\ \times_{\text{loop}} & : & a_K\mathtt{S}_i\mathtt{r}_i & \leq & a_0\mathtt{S}_i \\ \times_{\text{out}} & : & a_K\mathtt{S}_i\mathtt{S}_{i+1} & \leq & t_0\mathtt{S}_i\mathtt{S}_{i+1} \end{array} .$$

Set $\leq_{\times_i(q_{\text{in}}, q_{\text{out}})}$ to be the computation relation on the multiply by K program, and will write $\leq_{\times_i}$ when the program is understood in context.

We define the *subtract-K program*, denoted by $-_iK = (\mathtt{R}_3, \mathtt{Q}_{-_iK}, \mathtt{P}_{-_iK})$, for $i \in \{1, 2\}$, which is meant to subtract K tokens to register $\mathtt{r}_i$ and output state s_K. We define

$\mathtt{Q}_{-_i} = \{s_0, \ldots, s_K\}$ and the instructions $\mathtt{P}_{-_i K} = \{-_1, \ldots, -_K\}$, where:

$$\begin{array}{rcrcl} -_1 & : & s_0\mathtt{S}_i\mathtt{r}_i & \leq & s_1\mathtt{S}_i \\ -_2 & : & s_1\mathtt{S}_i\mathtt{r}_i & \leq & s_2\mathtt{S}_i \\ \vdots & & & \vdots & \\ -_K & : & s_{K-1}\mathtt{S}_i\mathtt{r}_i & \leq & s_K\mathtt{S}_i \end{array} .$$

We denote the computation relation on the subtract-K program by $\leq_-$.

We are now ready to define the *divide by* K program for $i \in \{1, 2\}$, denoted by $\div_i(q_{\text{in}}, q_{\text{out}}) = (\mathtt{R}_3, \mathtt{Q}_{\div_i}, \mathtt{P}_{\div_i}(q_{\text{in}}, q_{\text{out}}))$. We define the set $\mathtt{Q}_{\div_i} = \mathtt{Q}_{-_i} \cup \mathtt{Q}_{\mathtt{T}_i}$ and the instructions $\mathtt{P}_{\div_i}(q_{\text{in}}, q_{\text{out}}) = \mathtt{P}_{T_i}(q_{\text{out}}) \cup \mathtt{P}_{-_i K} \cup \{\div_{\text{in}}, \div_{\text{loop}}, \div_{\text{out}}\}$, where:

$$\begin{array}{rcrcl} \div_{\text{in}} & : & q_{\text{in}}\mathtt{S}_3\mathtt{S}_4 & \leq & s_0\mathtt{S}_3\mathtt{S}_4 \\ \div_{\text{loop}} & : & s_K\mathtt{S}_3 & \leq & s_0\mathtt{S}_3\mathtt{r}_3 \\ \div_{\text{out}} & : & s_0\mathtt{S}_i\mathtt{S}_{i+1} & \leq & t_0\mathtt{S}_i\mathtt{S}_{i+1} \end{array} .$$

We denote the computation relation on the divide by K program by $\leq_{\div_i(q_{\text{in}}, q_{\text{out}})}$, but will write $\leq_{\div_i}$ when understood in context.

Lastly, we define the *end program*, denoted by $F = (\mathtt{R}_3, \mathtt{Q}_F, \mathtt{P}_F)$ to be a transition of the final state q_f of $\mathtt{M}$ to the final state q_F of $\mathtt{M}_K$. We define $\mathtt{Q}_F = \{c_F, q_F\}$ and the instructions $\mathtt{P}_F = \{F_1, F_2\}$ are the following pair:

$$\begin{array}{rcrcl} F_1 & : & q_f\mathtt{S}_1\mathtt{r}_1 & \leq & c_F\mathtt{S}_i \\ F_2 & : & c_F\mathtt{S}_2\mathtt{r}_2 & \leq & q_F\mathtt{S}_2 \end{array} .$$

We define $\leq_F$ to be the computation relation on the end program.

We can now formally define the $\mathtt{M}_K$ machine. For a 2-CM $\mathtt{M} = (\mathtt{R}_2, \mathtt{Q}, \mathtt{P})$, define $\mathtt{P}_+$, $\mathtt{P}_-$, and $\mathtt{P}_\emptyset$ to be the sets of increment, decrement, and zero-test instructions, respectively,

from $\mathtt{P}$. Hence $\mathtt{P} = \mathtt{P}_+ \cup \mathtt{P}_- \cup \mathtt{P}_\emptyset$ is a disjoint union. Assume $\mathtt{Q}_F$ is disjoint from $\mathtt{Q}$, and for each increment and decrement instruction $p_+ : q_{\text{in}}\mathtt{S}_i \leq q_{\text{out}}\mathtt{S}_i\mathtt{r}_i \in \mathtt{P}_+$ and $p_- : q_{\text{in}}\mathtt{S}_i\mathtt{r}_i \leq q_{\text{out}}\mathtt{S}_i \in \mathtt{P}_-$, for $i \in \{1, 2\}$, we relabel the elements of the following sets

$$\mathtt{Q}_{\times_i}^{p_+} := \{q^{p_+} : q \in \mathtt{Q}_{\times_i}\} \quad \& \quad \mathtt{Q}_{\div_i}^{p_-} := \{q^{p_-} : q \in \mathtt{Q}_{\div_i}\}$$
$$\mathtt{P}_{\times_i}^{p_+} := \{p^{p_+} : p \in \mathtt{P}_{\times_i}(q_{\text{in}}, q_{\text{out}})\} \quad \& \quad \mathtt{P}_{\div_i}^{p_-} := \{p^{p_-} : p \in \mathtt{P}_{\div}(q_{\text{in}}, q_{\text{out}})\},$$

making the sets disjoint. Lastly, for each $p_\emptyset : q_{\text{in}}\mathtt{S}_i\mathtt{S}_{i+1} \leq q_{\text{out}}\mathtt{S}_i\mathtt{S}_{i+1} \in \mathtt{P}_\emptyset$, by Equation (5.1) we define $\mathtt{1}^{p_\emptyset} : q_{\text{in}}\mathtt{S}_i\mathtt{r}_i\mathtt{S}_{i+1} \leq q_{\text{out}}\mathtt{S}_i\mathtt{r}_i\mathtt{S}_{i+1}$, and

$$\mathtt{P}_1 := \{\mathtt{1}^{p_\emptyset} : p_\emptyset \in \mathtt{P}_\emptyset\}.$$

Definition 5.2.1. Let $\mathtt{M} = (\mathtt{R}_2, \mathtt{Q}, \mathtt{P})$ be a 2-CM and fix $K > 1$. We define the machine $\mathtt{M}_K := (\mathtt{R}_3, \mathtt{Q}_K, \mathtt{P}_K)$, where

- $\mathtt{Q}_K := \mathtt{Q} \cup \mathtt{Q}_F \cup \bigcup_{p \in \mathtt{P}_+} \mathtt{Q}_\times^p \cup \bigcup_{p \in \mathtt{P}_-} \mathtt{Q}_\div^p$,
- $\mathtt{P}_K := \mathtt{P}_1 \cup \mathtt{P}_F \cup \bigcup_{p \in \mathtt{P}_+} \mathtt{P}_\times^p \cup \bigcup_{p \in \mathtt{P}_-} \mathtt{P}_\div^p$.

Lemma 5.2.1. Let $p \in \mathtt{P}$ be an instruction acting on register $\mathtt{r}_i$, where $\{i, j\} = \{1, 2\}$. Let $\mathtt{C} = \langle q; n_1, n_2, n_3\rangle \in \mathtt{Conf}(\mathtt{M}_K)$ and suppose $\mathtt{C} \in \mathrm{Acc}(\mathtt{M}_K)$ witnessed by

$$\mathtt{C} \leq^{p_1} \mathtt{C}_1 \leq^{p_2} \cdots \leq^{p_N} \mathtt{C}_N = \mathtt{C}_F \in \mathrm{Acc}(\mathtt{M}_K).$$

1. If $p \in \mathtt{P}_+$ and $q \in \mathtt{Q}_\times^p$, then there exists $k \leq N$ such that $\mathtt{C} \leq_\times \mathtt{D} = \langle q_{\text{out}}; m_1, m_2, 0\rangle$, where $m_j = n_j$ and

   (i) $m_i = n_i + n_3 + \delta$, if $q = t_\delta^p$ for $\delta \in \{0, 1\}$;

   (ii) $m_i = Kn_i + n_3 + K - \delta$, if $q = a_\delta^p$ for $0 \leq \delta \leq K$.

2. If $p \in \mathtt{P}_{-}$ and $q \in \mathtt{Q}^{p}_{\div}$, then there exists $k \leq N$ such that $\mathtt{C} \leq_{\div} \mathtt{D} = \langle q_{\text{out}}; m_1, m_2, 0\rangle$, where $m_j = n_j$ and

   (i) $m_i = n_i + n_3 + \delta$, if $q = t^p_\delta$ for $\delta \in \{0, 1\}$;

   (ii) $K \mid (n_i + K - \delta)$ and $m_i = n_3 + \frac{n_i + K - \delta}{K}$, if $q = s^p_\delta$ for $0 \leq \delta \leq K$.

3. If $p \in \mathtt{P}_{\emptyset}$, then $\mathtt{C} \leq^{1^p} \langle q_{\text{out}}; m_1, m_2, n_3\rangle$ iff $m_j = n_j$ and $m_i = n_i = 1$.

Proof. The proofs of (1) and (2) are identically those given in Lemma 5.1.7, where only the arguments for the zero-test program are replaced by the same argument that $\mathtt{C} \leq^p \mathtt{C}'$ iff only if the i-th register of both $\mathtt{C}$ and $\mathtt{C}'$ are empty, where $p : q_{\text{in}}\mathtt{S}_i\mathtt{S}_{i+1} \leq q_{\text{out}}\mathtt{S}_i\mathtt{S}_{i+1}$ is an instruction internal to some multiply or divide program. (3) clearly holds by the definition of $\leq^{1^p}$ as a $\{\cdot\}$-compatible relation. □

Lemma 5.2.2. Let $\mathtt{C} = q_{\text{in}}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \in \mathtt{Conf}(\mathtt{M}_K)$ and suppose $q_{\text{in}} \in \mathtt{Q}$. Then $\mathtt{C} \in \text{Acc}(\mathtt{M}_K)$ if and only if there exists $\mathtt{C}' \in \mathtt{Conf}(\mathtt{M})$ such that $\mathtt{C}' \in \text{Acc}(\mathtt{M})$ and $\mathtt{C} = \mathtt{C}'_K$.

Proof. This proof is essentially the same as Lemma 5.1.8, where only the instructions corresponding sets $\mathtt{P}_{\emptyset}$ and $\mathtt{P}_1$ need to be checked. Consequently, it is sufficient to verify that for all $p \in \mathtt{P}_{\emptyset}$,

$$\mathtt{C} \leq^{1^p} \mathtt{D} \in \text{Acc}(\mathtt{M}_K) \iff (\exists \mathtt{C}', \mathtt{D}' \in \mathtt{Conf}(\mathtt{M}))\ \mathtt{C}' \leq^p \mathtt{D}' \in \text{Acc}(\mathtt{M}) \text{ and } \mathtt{C} = \mathtt{C}'_K, \mathtt{D} = \mathtt{D}'_K.$$

Let $p : q_{\text{in}}\mathtt{S}_i\mathtt{S}_{i+1} \leq q_{\text{out}}\mathtt{S}_i\mathtt{S}_{i+1}$ for some $i \in \{1, 2\}$ and $q_{\text{in}}, q_{\text{out}} \in \mathtt{Q}$. Since $q_F \notin \mathtt{Q}$, $q_{\text{out}} \neq q_F$. So by the construction of $\mathtt{M}_K$ and the definition of the end program, $q_{\text{out}} \in \mathtt{Q}$, $\mathtt{C} \in \text{Acc}(\mathtt{M}_K)$ if and only if $\mathtt{C} \leq_{\mathtt{M}_K} q_f\mathtt{S}_1\mathtt{r}_1\mathtt{S}_2\mathtt{r}_2\mathtt{S}_3\mathtt{S}_4 = (\mathtt{C}_f)_K$, where $\mathtt{C}_f$ is the final configuration for $\mathtt{M}$. This observation completes the base case for induction for each direction. The inductive steps follow from Lemma 5.1.7(3).

□

Let $\tilde{\mathtt{M}}$ be the 2-CM given by Theorem 4.2.1. Since membership of $\mathrm{Acc}(\tilde{\mathtt{M}})$ is undecidable, we obtain the following:

Corollary 5.2.3. Membership of the set $\mathrm{Acc}(\tilde{\mathtt{M}}_K)$ is undecidable for $K > 1$.

5.2.2 Simple equations and admissibility for CMs.

Consider a simple equation $[\mathrm{D}] \in \mathcal{U}$. By Theorem 5.3.1, the word problem for $\mathsf{RL}+[\mathrm{D}]$ is undecidable, witnessed in its $\{\vee, \cdot, 1\}$-fragment. We may inquire whether the root of undecidability can be traced further down to its $\{\leq, \cdot, 1\}$-fragment in the same way by using the machine $\tilde{\mathtt{M}}_K$. However, unlike Section 5.1.3, a more delicate approach is necessary to prove admissibility. This technicality is rooted in that many rules in $\mathcal{U}$ may have instances that allow stopper variables to permute amongst register terms, potentially allowing non-configurations to be accepted. Since our machines are $\{\mathtt{S}_i\}$-stable, if there are instances of $\leq^{\mathrm{D}}$ that have this effect, then it implies there exists $x \in \mathrm{supp}(\mathrm{D})$ such that $\#(d, x) = 1$ and $d = u_d x v_d$ where $x \notin \mathrm{supp}(u_d v_d)$, for all $d \in \mathrm{D}$. Therefore, if $[\mathrm{D}] \in \mathcal{U}$ is such that, for all $x \in \mathrm{supp}(\mathrm{D})$ there exists $d \in \mathrm{D}$ such that $\#(d, x) \neq 1$, then $[\mathrm{D}]$ is strongly admissible in $\mathtt{M}_K$ by essentially the same argument as Theorem 5.3.1, establishing undecidability of the $\{\leq, \cdot, 1\}$-fragment.

We first establish the following technical lemma as a consequence of Lemma 5.1.11.

Lemma 5.2.4. Let $[\mathrm{D}] \in \mathcal{U}$ be a simple equation and $\mathtt{M}$ a 2-CM. Then for all sufficiently large $K > 1$, if $\mathtt{C} \in \mathtt{Conf}(\mathtt{M}_K)$ is such that $\mathtt{C} \leq^{\mathrm{D}} v \in \mathrm{Acc}(\mathtt{M}_K)$ then $\mathtt{C} \in \mathrm{Acc}(\mathtt{M})$.

Proof. Let $[\mathrm{D}] = (\mathbf{1_n}, \mathrm{D}) \in \mathcal{U}$. By Theorem 5.3.1, for all K sufficiently large $[\mathrm{D}], K \models (\star\star)$. Fix such a $K > 1$. Let $\mathtt{C} \in \mathtt{Conf}(\mathtt{M}_K)$ and suppose $\mathtt{C} \leq^{\mathrm{D}} v \in \mathrm{Acc}(\mathtt{M}_K)$. Since $\mathtt{Conf}(\mathtt{M}_K)$ is stable in $\mathtt{M}_K$ by Lemma 4.2.2, $v = \bigvee_{d \in \mathrm{D}} \mathtt{C}_d \in \mathtt{Conf}(\mathtt{M}_K)^{\vee}$. Hence $\mathtt{C}_d \in \mathrm{Acc}(\mathtt{M}_K)$ for each $d \in \mathrm{D}$ by Proposition 4.1.12. Let $\boldsymbol{\sigma}$ be the substitution such that

$$\mathtt{C} = u\boldsymbol{\sigma}(\mathbf{1_n})v \leq^{\mathrm{D}} \bigvee_{d \in \mathrm{D}} u\boldsymbol{\sigma}(d)v,$$

where u, v are monoid terms and $\mathtt{C}_d = u\boldsymbol{\sigma}(d)v$ for each $d \in \mathrm{D}$. For each $d \in \mathrm{D}$, define $\overline{d} \in \mathbb{N}^n$ via $\overline{d}(i) := \#(d, x_i)$, where $\{x_1, \ldots, x_n\}$ is set of distinct variables appearing in $[\mathrm{D}]$. Under this notation, $\overline{\mathbf{1_n}} = (1, \ldots, 1) \in \mathbb{N}^n$. For each $j \in \{1, 2, 3\}$, define $\sigma_i \in (\mathbb{N}^n)^T$ via $\sigma_j(i) = \#(\boldsymbol{\sigma}(\mathbf{1_n}), \mathtt{r}_i)$ and $C_j \in \mathbb{N}$ via $C_j = \#(uv, \mathtt{r}_i)$. Therefore there is a $q \in \mathtt{Q}_K$ such that

$$\begin{array}{rcl} \mathtt{C} & = & \langle q; C_1 + \sigma_1\overline{\mathbf{1_n}}, C_2 + \sigma_2\overline{\mathbf{1_n}}, C_2 + \sigma_2\overline{\mathbf{1_n}}\rangle \\ \mathtt{C}_d & = & \langle q; C_1 + \sigma_1\overline{d}, C_2 + \sigma_2\overline{d}, C_2 + \sigma_2\overline{d}\rangle \end{array},$$

for each $d \in \mathrm{D}$. This is precisely the same setup as Lemma 5.1.11, and thus by following the same arguments and using the corresponding Lemma 5.2.1, we deduce that $\mathtt{C} \in \mathrm{Acc}(\mathtt{M}_K)$. □

Motivated by above, let $\mathcal{U}_{-1}$ be the class of all $[\mathrm{D}] = (\mathbf{1_n}, \mathrm{D}) \in \mathcal{U}$ such that for every $i = 1, \ldots, n$ there exists $d_i \in \mathrm{D}$ such that $\#(d, x_i) \neq 1$. It is straightforward to verify, in the style of Theorem 2.4.1, that $\Sigma \subseteq \mathcal{U}_{-1}$ implies Σ does not entail commutativity (see Section 4.2.2).

Lemma 5.2.5. Let $\Sigma \in \mathcal{U}_{-1}$ be finite and $\mathtt{M}$ a 2-CM. Then there exists $K > 1$ such that $[\Sigma]$ is strongly admissible in $\mathtt{M}_K$.

Proof. Since Σ is finite, by Corollary 5.3.14 there exists a $K > 1$ such that $\Sigma, K \models (\star\star)$. Fix $[\mathrm{D}] = (\mathbf{1_n}, \mathrm{D}) \in \Sigma$. By Lemma 4.1.11, it is enough to show that if $t \leq^{\mathrm{D}} \bigvee_{d\in\mathrm{D}} t_d$ for some monoid terms t and $\{t_d : d \in \mathrm{D}\}$, then

$$\forall d \in \mathrm{D}, t_d \in \mathrm{Acc}(\mathtt{M}_K) \implies t \in \mathrm{Acc}(\mathtt{M}_K).$$

Now, $t_d \in \mathrm{Acc}(\mathtt{M}_K)$ implies $t_d = \mathtt{C}_d \in \mathtt{Conf}(\mathtt{M}_K)$ by Lemma 4.2.2. So $t \leq^{\mathrm{D}} \bigvee_{d\in\mathrm{D}} \mathtt{C}_d$. We first show that $t \in \mathtt{Conf}(\mathtt{M}_K)$.

Since Σ does not entail commutativity, it follows that t must be of the form Equation (4.7) from Lemma 4.2.5, i.e., $t = uqv$ for some $q \in \mathtt{Q}_K$ and monoid terms u, v such that $uv = \mathtt{S}_1 x_1 \mathtt{S}_2 x_2 \mathtt{S}_3 x_3 \mathtt{S}_4$ for some $x_1, x_2, x_3 \in \mathtt{R}_3^*$. We wish to show $uv \in \mathtt{Box}(\mathtt{M}_K)$.

Let $\boldsymbol{\sigma}$ be the substitution witnessing $t \leq^{\mathrm{D}} \bigvee_{d \in \mathrm{D}} \mathtt{C}_d$, i.e.,

$$uv = w\boldsymbol{\sigma}(\mathbf{1_n})w' \leq^{\mathrm{R}} \bigvee_{d \in \mathrm{D}} w\boldsymbol{\sigma}(d)w',$$

where w, w' are monoid words. Since each $\mathtt{C}_d =_{\mathtt{Q}_K} qw\boldsymbol{\sigma}(d)w'$ is a configuration and $[\mathrm{D}] \in \mathcal{U}_{-1}$, by definition it follows that no variable from $\mathtt{Q}_K \cup \mathtt{Stp}_3$ that is a subword of $\boldsymbol{\sigma}(d)$, for all $d \in \mathrm{D}$. This implies that

$$ww' = \mathtt{S}_1 \mathtt{r}_1^{m_1} \mathtt{S}_2 \mathtt{r}_2^{m_2} \mathtt{S}_3 \mathtt{r}_3^{m_3} \mathtt{S}_4 \in \mathtt{Box}(\mathtt{M}_K)$$

and there is an $i \in \{1, 2, 3\}$ such that for each $d \in \mathrm{D}$, $\boldsymbol{\sigma}(d) = \mathtt{r}_i^{m_d}$ since $w\boldsymbol{\sigma}(d)w' \in \mathtt{Box}(\mathtt{M}_K)$. Since $[\mathrm{D}] \in \mathcal{U}$, $[\mathrm{D}]$ is not integral and hence $\boldsymbol{\sigma}(\mathbf{1_n}) = \mathtt{r}_i^m$. Hence, $uv = w\boldsymbol{\sigma}(\mathbf{1_n})w' \in \mathtt{Box}(\mathtt{M}_K)$. Therefore $t \in \mathtt{Conf}(\mathtt{M}_K)$.

Since $\Sigma, K \models (\star\star)$ and $t \in \mathtt{Conf}(\mathtt{M}_K)$, by Lemma 5.2.4, we obtain $t \in \mathrm{Acc}(\mathtt{M}_K)$. Therefore Σ is strongly admissible in $\mathtt{M}_K$. □

Therefore, by Corollary 5.2.3 and Corollary 4.1.10 we obtain:

Theorem 5.2.6. Let $\Sigma \subseteq \mathcal{U}_{-1}$ be finite. Then the $\{\leq, \cdot, 1\}$-fragment of the word problem for $\mathsf{RL} + \Sigma$ is undecidable

We note that the above is only a sufficient condition for our result. One can define weaker conditions that imply canonical admissibility. However, we will only motivate such an investigation with the following example.

Example 5.2.1. Consider the simple equation $[\mathrm{R}_{\rightarrow}]$ given by

$$[\mathrm{R}_{\rightarrow}] : xy \leq yx \vee y.$$

Now, $[\mathrm{R}_{\rightarrow}]_{\mathrm{com}} : xy \leq xy \vee y$, which is trivial and therefore $[\mathrm{R}_{\rightarrow}] \in \mathcal{U}$ but $[\mathrm{R}_{\rightarrow}] \notin \mathcal{U}_{-1}$. Consider the machine $\mathtt{M}_K$ where $\mathtt{M} = \mathtt{M}_{\mathrm{even}}$ and $K > 1$. Recall that for a monoid term t, $t \in \mathrm{Acc}(\mathtt{M}_K)$ implies $t \in \mathtt{Conf}(\mathtt{M}_K)$. Hence, for $q_0 \in \mathtt{Q}_{\mathrm{even}}$ and $u, v \in A_{\mathtt{M}_K}$, we observe that $uq_0v \in \mathrm{Acc}(\mathtt{M}_K)$ iff $uq_0v \in \mathtt{Conf}(\mathtt{M}_K)$ and $uq_0v = \langle q_0; K^{2n} \rangle$. Consider the substitution that maps $x \mapsto \mathtt{r}_1^{K^2-1}$ and $y \mapsto \mathtt{S}_1$. Then $\mathtt{r}_1^{K^2-1}\mathtt{S}_1 \leq \mathtt{S}_1\mathtt{r}_1^{K^2} \vee \mathtt{S}_1$, and therefore

$$q_0\mathtt{r}_1^{K^2-1}\mathtt{S}_1\mathtt{r}_1\mathtt{S}_2 \leq^{\mathrm{R}_{\rightarrow}} q_0\mathtt{S}_1\mathtt{r}_1^{K^2}\mathtt{S}_2 \vee q_0\mathtt{S}_1\mathtt{r}_1\mathtt{S}_2 \in \mathrm{Acc}(\mathtt{M}_K),$$

since in both joinands the $\mathtt{r}_1$-register contains K^2 and K^0 many tokens, respectively. However, $q_0\mathtt{r}_1^{K^2-1}\mathtt{S}_1\mathtt{r}_1\mathtt{S}_2 \notin \mathtt{Conf}(\mathtt{M}_K)$, and therefore $[\mathrm{R}_{\rightarrow}]$ is not strictly admissible in $\mathtt{M}_K$.

Although strict admissibility fails for $[\mathrm{R}_{\rightarrow}]$, since we can only permute variables in one direction, we will prove that $[\mathrm{R}_{\rightarrow}]$ is admissible in $\mathtt{M}_K$, i.e., for every $\mathtt{C} \in \mathtt{Conf}(\mathtt{M}_K)$,

$$\mathtt{C} \in \mathrm{Acc}(\mathrm{R}_{\rightarrow}\mathtt{M}_K) \iff \mathtt{C} \in \mathrm{Acc}(\mathtt{M}_K).$$

Roughly, the argument is as follows: The only way an instance of $\leq^{\mathrm{R}_{\rightarrow}}$ that, when applied to a configuration, results in a non-configuration is if a $\mathtt{r}_1$-variable permutes over the stopper $\mathtt{S}_2$. Since there are no instances in $\leq_{\mathtt{M}_K}$ nor instances of $\leq^{\mathrm{R}_{\rightarrow}}$ that can "undo" this effect, such an application cannot result to an accepted term in $\mathtt{M}_K$.

5.3 Membership of $\mathcal{U}$

5.3.1 The class of equations $\mathcal{U}$. We will now define a class $\mathcal{U} \subseteq \mathcal{S}$ of simple equations for which we will show $(\mathsf{C})\mathsf{RL} + [\mathrm{D}]$ has an undecidable word problem, for $[\mathrm{D}] \in \mathcal{U}$. The

collection $\mathcal{U}$ is so vast that it is easier to define its complement in $\mathcal{S}$. We motivate the definition with the following observation.

Consider the machine $\mathtt{B}_{\text{even}}$ and the simple equation $[\mathrm{R}] : x \leq x^2 \vee 1$. As before, it is easy to see that $q_0\mathtt{r}_1^3 \in \mathrm{Acc}(\mathrm{R}\mathtt{B}_{\text{even}}) \setminus \mathrm{Acc}(\mathtt{B}_{\text{even}})$. However, this behavior cannot be controlled with $\mathtt{B}_K$ for any $K > 1$. E.g., let $n = (K^4 - K^2)/2$, then $q_0\mathtt{r}_1^{K^2+n} \notin \mathrm{Acc}(\mathtt{B}_K)$ since $K^2 + n \neq K^{2m}$ for any $m \in \mathbb{N}$, however

$$q_0\mathtt{r}_1^{K^2+n} = q_0\mathtt{r}_1^{K^2}\mathtt{r}_1^n \leq^{\mathrm{R}} q_0\mathtt{r}_1^{K^2}\mathtt{r}_1^{2n} \vee q_0\mathtt{r}_1^{K^2}\mathtt{r}_1^0 = q_0\mathtt{r}_1^{K^4} \vee q_0\mathtt{r}_1^{K^2} \in \mathrm{Acc}(\mathtt{B}_K).$$

In fact, we will show this failure occurs, not just for functions of the form $n \mapsto K^n$ but actually for any (computable) injective function on $\mathbb{N}$.

Given the natural ordering of our variable set $\{x_i : i \in \mathbb{Z}^+\}$, note that using our vector notation, every commutative monoid term can be written in the form $\mathbf{x_n}^f$, for some $n \in \mathbb{Z}^+$ and f an n-tuple of natural numbers; recall that $\mathbf{x_n} = (x_1, \ldots, x_n)$. If we actually extend our notation to the case where $\mathbf{x}_\infty = (x_i)_{i\in\mathbb{Z}^+} = (x_1, x_2, \ldots)$ and f is a sequence of natural numbers that is eventually constantly zero, then every commutative monoid term is of the form $\mathbf{x}_\infty{}^f$, and thus it is fully specified by such an f. In the following we will work interchangeably in the free monoid over the variable set $\{x_i : i \in \mathbb{Z}^+\}$ and also in the isomorphic monoid $\mathbb{F}$ of eventually-zero sequences of natural numbers. More formally, $\mathbb{N}^{\mathbb{Z}^+}$ denotes the set of all functions from $\mathbb{Z}^+$ to $\mathbb{N}$ and for $f \in \mathbb{N}^{\mathbb{Z}^+}$, we define $\mathrm{supp}(f) := \{i \in \mathbb{Z}^+ : f(i) \neq 0\}$ to be the *support* of f. Then the set $\mathbb{F} := \{f \in \mathbb{N}^{\mathbb{Z}^+} : |\mathrm{supp}(f)| < \infty\}$ of all functions of finite support forms a commutative monoid $(\mathbb{F}, +, \mathbf{0})$, under addition and with unit the constantly-zero function $\mathbf{0}$. Clearly, this monoid is simply an additive rendering of the free commutative monoid on countably many generators and isomorphic to the above multiplicative rendering by exactly the map $f \mapsto \mathbf{x}_\infty{}^f$, and we will freely

move between the two representations. Under this isomorphism the variable x_i maps to the generator $\mathbf{e}_i$, which has 1 in the i-th entry and 0 everywhere else.

For reasons that will be clear soon, we view the elements of $\mathbb{F}$ as column vectors and we also consider the bijective set $\mathbb{F}^T$ of the row vectors, which are the transposes of the elements of $\mathbb{F}$. In particular, for $f \in \mathbb{F}$ and $\sigma \in \mathbb{F}^T$, the matrix product σf yields a 1×1 matrix, which we identify with the natural number equal to its unique entry. Even though f and σ are each of infinite dimension, they both have finite support, so their product is well defined. For a set $X \subseteq \mathbb{F}$, we write $\sigma X := \{\sigma f \in \mathbb{N} : f \in X\}$ and $\mathrm{supp}(X) := \bigcup_{f \in X} \mathrm{supp}(f)$. For $n \in \mathbb{N}$, we will often define the set $\mathbf{n} := \{1, \ldots, n\}$ for ease of notation.

In this way, for $n \in \mathbb{Z}^+$, the *n-variable linear vector* $\mathbf{1_n} \in \mathbb{F}$ is written

$$\mathbf{1_n} := \sum_{k \in \mathbf{n}} \mathbf{e}_k.$$

Definition 5.3.1. We identify the proper ISR-equations for CRL by the set $\mathcal{A} \subseteq \mathbb{F} \times \wp(\mathbb{F})$, where $(a_0, A) \in \mathcal{A}$ if and only if A is finite and $\mathrm{supp}(A) \subseteq \mathrm{supp}(a_0)$. Similarly, the simple equations for CRL by are represented by $\mathcal{S} \subseteq \mathcal{A}$, where $(a_0, A) \in \mathcal{S}$ if $a_0 = \mathbf{1_n}$ for some $n \in \mathbb{Z}^+$.

We see that each proper n-variable equation $[A]$ corresponds to some $(a_0, A) \in \mathcal{A}$ via

$$\mathbf{x_n}^{a_0} \leq \bigvee_{r \in A} \mathbf{x_n}^{r}.$$

By the terminology of Definition 2.1.1, an equation $[A] = (a_0, A)$ is *trivial* if $a_0 \in A$, and *integral* if $\operatorname{supp}(A) \subsetneq \operatorname{supp}(a_0)$.[5]

Figure 5.1 contains examples of simple equations $[\mathrm{R}]$ viewed as sets of vectors R:

	$[\mathrm{R}]$	$\mathbf{1_n}$	R
(i)	$x \leq 1$	$\mathbf{1_1}$	$\{\mathbf{0}\}$
(ii)	$x \leq x^2$	$\mathbf{1_1}$	$\{2\mathbf{e}_1\}$
(iii)	$x \leq x^2 \vee 1$	$\mathbf{1_1}$	$\{2\mathbf{e}_1, \mathbf{0}\}$
(iv)	$x \leq x^2 \vee x^4$	$\mathbf{1_1}$	$\{2\mathbf{e}_1, 4\mathbf{e}_1\}$
(v)	$xy \leq x^2y \vee x^3y^2$	$\mathbf{1_2}$	$\{2\mathbf{e}_1 + \mathbf{e}_2, 3\mathbf{e}_1 + 2\mathbf{e}_2\}$
(vi)	$xyz \leq x^2y \vee y^2z \vee xz^2$	$\mathbf{1_3}$	$\{2\mathbf{e}_1 + \mathbf{e}_2, 2\mathbf{e}_2 + \mathbf{e}_3, \mathbf{e}_1 + 2\mathbf{e}_3\}$
(vii)	$xyz \leq xz^2 \vee yz$	$\mathbf{1_3}$	$\{\mathbf{e}_1 + 2\mathbf{e}_3, \mathbf{e}_2 + \mathbf{e}_3\}$

Figure 5.1: Simple equations as set of vectors

A substitution $\boldsymbol{\sigma}$ on $\mathbb{F}$ is fully determined by its application on the generators $\mathbf{e}_i \mapsto f_i \in \mathbb{F}$ for each $i \in \mathbb{Z}^+$, and as it is a homomorphism, namely an additive/linear map, its application is given by multiplication of an associated matrix $M_{\boldsymbol{\sigma}}$; so $\boldsymbol{\sigma}(f) = M_{\boldsymbol{\sigma}} f$. Since we only consider finite sets $\mathrm{R} \subseteq \mathbb{F}$ for equations $[A] = (a_0, A) \in \mathcal{S}$, we may view $\mathrm{R} \subseteq \mathbb{N}^n$ and, in this way, will only consider substitutions $\boldsymbol{\sigma} : \mathbb{N}^n \to \mathbb{N}^k$, in which case the associated $M_{\boldsymbol{\sigma}}$ is a $k \times n$ matrix; in this case, we say $\boldsymbol{\sigma}$ is a *k-variable substitution*. We will write $\sigma_i \in \mathbb{N}^n$ for the i-th row of $M_{\boldsymbol{\sigma}}$ for each $i \leq k$ and also $M_{\boldsymbol{\sigma}} = (\sigma_i)_{i=1}^k$. Abusing notation, we will identify $\boldsymbol{\sigma} = M_{\boldsymbol{\sigma}} = (\sigma_i)_{i=1}^k$. If $[A] = (a_0, A) \in \mathcal{A}$, then $(\boldsymbol{\sigma} a_0, \boldsymbol{\sigma} A) \in \mathcal{A}$, which we denote by $[\boldsymbol{\sigma} A]$.

Definition 5.3.2. We say a finite set $V \subseteq \mathbb{F}$ is a *spine* if $V = \{\mathbf{0}\}$ or $V \setminus \{\mathbf{0}\} = \{v_1, \ldots, v_k\}$ such that $i \in \operatorname{supp}(v_i) \subseteq \{1, \ldots, i\}$ for each $1 \leq i \leq k$. If $\boldsymbol{\sigma}$ is a substitution, we say V is a *$\boldsymbol{\sigma}$-spine* if $\boldsymbol{\sigma} V$ is a spine. We say an equation $[V] = (f, V) \in \mathcal{A}$ is *spinal* if $[V]$ is

[5]That is, $a_0 \in A$ implies $\mathsf{RL} \models [A]$, and by Proposition 2.1.2, $\operatorname{supp}(a_0) \setminus \operatorname{supp}(A) \neq \emptyset$ implies $\mathbf{R} \models [A] \iff \mathbf{R} \models x \leq 1$ for any $\mathbf{R} \in \mathsf{RL}$.

nontrivial and V is a spine. I.e., viewing it as an ISR-equation, $[V]$ is equivalently written as:

$$[V] : \underbrace{x_1^{f(1)} \cdots x_k^{f(k)}}_{f} \leq \underbrace{(1 \vee)}_{\mathbf{0}} \underbrace{x_1^{v_1(1)}}_{v_1} \vee \underbrace{x_1^{v_2(1)} x_2^{v_2(2)}}_{v_2} \vee \cdots \vee \underbrace{x_1^{v_k(1)} \cdots x_k^{v_k(k)}}_{v_k},$$

where $(1\vee)$ is meant to signify 1 may or may not be included in the join, i.e., whether $\mathbf{0}$ is contained in V. We say a simple equation $[\mathrm{R}] = (\mathbf{1_n}, \mathrm{R}) \in \mathcal{S}$ is *pre-spinal* if there is a substitution $\boldsymbol{\sigma}$ such that $[\boldsymbol{\sigma}\mathrm{R}]$ is spinal.

Note that all knotted equations $[\mathsf{k}_n^m] : x^n \leq x^m$ are spinal and so their equivalent simple equations $[\mathsf{K}_n^m]$ (as defined in Section 2.4) are pre-spinal. As a consequence of the definition, $[\mathrm{R}]$ is spinal if an only if $[\mathrm{R} \cup \{\mathbf{0}\}]$ is spinal, so all equations $x^n \leq x^m \vee 1$ are spinal as well.[6] From Table 5.1, we see that (i)-(iii) are spinal. The simple equation (vii) is pre-spinal via the 1-variable substitution $\boldsymbol{\sigma}$ given by $\boldsymbol{\sigma} := (\mathbf{e}_1 + \mathbf{e}_2)^T$, i.e., $\mathsf{CRL} + (vii) \models x^2 \leq x$. On the other hand, no trivial equations are pre-spinal. The general characterization of whether a simple equation is pre-spinal will be addressed in Section 5.3, where it can be verified that $(iv) - (vi)$ in Table (5.1) are not pre-spinal by Theorem 5.3.5.

Definition 5.3.3. The set $\mathcal{U}$ contains all simple equations that are not pre-spinal. If $[\mathrm{R}]$ is any simple equation in RL, we write $[\mathrm{R}] \in \mathcal{U}$ iff $[\mathrm{R}]_{\mathrm{com}} \in \mathcal{U}$.

In the following sections we will prove, in particular, the following theorem as a consequence of Lemma 4.1.11, Lemma 5.1.11, Corollary 5.1.9, and Corollary 5.3.14:

Theorem 5.3.1. Let $\Gamma \subseteq \mathcal{U}$ be finite. Then any variety $\mathcal{V}$ in the interval $\mathsf{CRL} + \Gamma \subseteq \mathcal{V} \subseteq \mathsf{RL}$ has an undecidable word problem.

In particular, by Corollary 4.3.10 we prove:

[6] It should be noted that knotted extensions of CRL have the FEP (see Proposition 3.1.1) and hence a decidable word problem, but decidability results for $x^n \leq x^m \vee 1$ are unknown to this author.

Theorem 5.3.2. Let $\Gamma \subseteq \mathcal{U}$ be finite. If $\mathsf{CRL} + \Gamma$ is expansive then it has an undecidable equational theory.

We will characterize the complement of $\mathcal{U}$ by giving conditions for when a simple equation $[\mathrm{D}]$ is pre-spinal. We will show that if there are infinitely many K such that $[\mathrm{D}], K \not\models (\star\star)$ then we can construct $\boldsymbol{\sigma}$ witnessing the pre-spinality of $[\mathrm{D}]$. Then by Lemma 5.3.4, $[\mathrm{D}] \in \mathcal{U}$ if and only if there is $N > 1$ where $[\mathrm{D}], K \models (\star\star)$ for all $K > N$, establishing Theorem 5.3.1. To that aim, we make the following definitions and observations.

For $f \in \mathbb{F}$ and for $S \subseteq \mathbb{Z}^+$ finite, we write $f[S]$ to be the restriction of f to the indices S, and can naturally view $f[S] \in \mathbb{N}^{|S|}$. We say f is *S-positive* if $f[S] > \mathbf{0}$, i.e., $f[S] \neq \mathbf{0}$ and $f(i) \geq 0$ for each $i \in S$. For $T \subseteq S$, we say f is *(T, S)-positive* if f is T-positive and $\mathrm{supp}(f) \subseteq S$. For $\mathrm{D} \subseteq \mathbb{F}$ with $\mathrm{supp}(\mathrm{D}) \subseteq \mathbf{n}$, we write $\mathrm{D}[S] = \{d[S] : d \in \mathrm{D}\}$ and will interchangeably view $\mathrm{D}[S]$ as both set or a $|S| \times |\mathrm{D}|$ matrix with columns from $\mathrm{D}[S] \subseteq \mathbb{N}^n$; in which case we denote the *i-th row of* D by $\mathrm{D}[i] := \mathrm{D}[\{i\}]$.

By definition, $[\mathrm{D}] = (\mathbf{1_n}, \mathrm{D})$ is pre-spinal iff there exists a substitution $\boldsymbol{\sigma}$ such that $\boldsymbol{\sigma}\mathrm{D} \subseteq \mathbb{F}$ is a spine and $\boldsymbol{\sigma}\mathbf{1_n} \notin \boldsymbol{\sigma}\mathrm{D}$, ensuring it is not trivial. Since no integral equation is in $\mathcal{U}$ by Lemma 5.1.10, we will assume $\mathrm{supp}(\mathrm{D}) = \mathbf{n}$ henceforth. Similarly, if $\boldsymbol{\sigma}\mathrm{D}$ is a spine then we will assume $\boldsymbol{\sigma}\mathrm{D} \setminus \{\mathbf{0}\} \neq \emptyset$. Since $(\mathbf{1_n}, \mathrm{D})$ is pre-spinal if and only if $(\mathbf{1_n}, \mathrm{D} \cup \{\mathbf{0}\})$ is pre-spinal, we will assume any spine is of the form $V = \{v_0, v_1, \ldots, v_k\}$, where always $v_0 = \mathbf{0}$ and v_i given so that $i \in \mathrm{supp}(v_i) \subseteq \{1, \ldots, i\}$ for each $1 \leq i \leq k$.

5.3.2 Spinal equations. Let $V \subset \mathbb{F}$ be a spine. Viewing it as a matrix of column vectors $[v_0\ v_1\ \cdots\ v_k]$, we see that V is an upper-right triangular matrix such that $v_i(i) \neq 0$. For $f \in \mathbb{F}$, $(f, V) \in \mathcal{A}$ only if $\mathrm{supp}(V) \subseteq \mathrm{supp}(f)$, and is furthermore spinal only if $f \notin V$. It easily follows that $[V] = (f, V) \in \mathcal{A}$ is spinal only if $f \neq v_k$. We say $[V]$ is *reduced-spinal* if furthermore $f(1) \neq v_k(1)$ but $f(i) = v(i)$ for all $i > 1$. So for every spinal equation $[V]$,

$(f[X], V[X])$ is reduced-spinal, where $m = \max\{i : f(i) \neq v_k(i)\}$ and $X = \{m, \ldots, k\}$. That is, $[\boldsymbol{\tau} V]$ is reduced-spinal, where $\boldsymbol{\tau} = (\mathbf{e}_{i+m}^T)_{i=0}^{k-m}$.

$$V = \left.\begin{bmatrix} 0 & v_1(1) & \cdots & v_m(1) & \cdots & v_k(1) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \hdashline 0 & 0 & \cdots & v_m(m) & \cdots & v_k(m) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & v_k(k) \end{bmatrix}\right\} \boldsymbol{\tau} V \quad ; \quad f = \left.\begin{bmatrix} f(1) \\ \vdots \\ \hdashline f(m) \\ \vdots \\ f(k) \end{bmatrix}\right\} \boldsymbol{\tau} f$$

Figure 5.2: Reduced-spinal equation

Lemma 5.3.3. Let $[V] = (f, V) \in \mathcal{A}$ be spinal. For any injection $\phi : \mathbb{N} \to \mathbb{N}$ there exists $\tau \in \mathbb{F}^T$ and $C \in \mathbb{N}$ such that $C + \tau V \subseteq \phi[\mathbb{N}]$ but $\tau f \notin \tau V$.

Proof. By the above observation, we may assume $[V]$ is reduced-spinal. By definition of $[V]$ being reduced, $f(1) \neq v_k(1)$ but $f(i) = v_k(i)$ for each $i > 1$. Hence $\tau f \neq \tau v_k$ for any $\tau \in \mathbb{F}^T$ with $\tau(1) > 0$. To ensure $\tau f \notin \tau V$, it is enough to construct a τ such that $\tau(i+1) > \tau v_i$ for each $i = 1, \ldots, k-1$, which is well defined since $v_i(i+j) = 0$ for any $j \geq 1$. Indeed, if any τ satisfying such a property, then for each $1 \leq i < k$,

$$\tau f = \sum_{j=1}^{k} \tau(j) f(j) \geq \tau(i+1) f(i+1) \geq \tau(i+1) > \tau v_i,$$

since $f(j) > 0$ for each $j = 1, \ldots, k$ by definition of $[V]$ being reduced-spinal. And hence $\tau f \notin \tau V$.

Let ϕ be an injection. With the above in mind, we will construct a $\tau \in \mathbb{F}^T$ satisfying the property that $\tau(i+1) > \tau v_i$ for each $i = 1, \ldots, k-1$, as well as ensuring $C + \tau V \subseteq \phi[\mathbb{N}]$ for some $C \in \mathbb{N}$.

Define $N_i := \prod_{j=i}^{k} v_j(j)$ for each $1 \leq i \leq k$. Since ϕ is an injection, $\phi[\mathbb{N}]$ is infinite, there exists an infinite subset $A \subseteq \mathbb{N}$ such that for all $a, b \in A$, $\phi(a) \equiv \phi(b) \mod N_1$. Define $C = \min\{\phi(a) : a \in A\}$ and $a_0 = \phi^{-1}(C)$. We inductively construct τ such

that $C + \tau v_i \in \phi[\mathbb{N}]$. For $n = 1$, since $\phi[A]$ is infinite, there exists $a_1 \in A$ such that $\phi(a_1) - C = N_1 k_1 = t_1 v_1(1)$, for some $k_1 > \phi(a_0) = C$ and $t_1 := N_2 k_1$. Hence $C + (t_1 \mathbf{e}_1^T) v_1 = C + t_1 v_1(1) = \phi(a_1)$. Define $\tau_1 = t_1 \mathbf{e}_1^T \in \mathbb{F}^T$.

Suppose that for $n \geq 1$, there exists $t_1, \ldots, t_n$ and $a_1, \ldots, a_n$ such that $C + \tau_n v_i = \phi(a_i)$, where $\tau_n = \sum_{i=1}^n t_i \mathbf{e}_i^T$, and $t_i = N_{i+1} k_i$ with $k_i > \phi(a_{i-1})$, for each $1 \leq i \leq n$. By definition for $a \in A$, there exists $m_a \in \mathbb{N}$ such that $\phi(a) - \phi(a_0) = N_1 m_a$. Let $x_a := \phi(a) - C - \tau_n v_{n+1} \in \mathbb{Z}$. By the induction hypothesis,

$$x_a = N_1 m_a - \sum_{i=1}^{n} N_{i+1} k_i v_{n+1}(i) \equiv 0 \mod N_{n+1}, \tag{5.2}$$

since $C = \phi(a_0)$ and $N_{n+1} \mid N_i$ for all $i \leq n+1$. Since $\phi[A]$ is infinite, there exists infinite $A' \subseteq A$ such that $x_a > 0$ for each $a \in A'$. So by Equation 5.2, there exists $\bar{a} \in A'$ such that $x_{\bar{a}} = N_{n+1} k_{\bar{a}} = t_{\bar{a}} v_{n+1}(n+1)$, for some $k_{\bar{a}} \geq \phi(a_n)$ and $t_{\bar{a}} := N_{n+2} k_{\bar{a}}$. We set $a_{n+1} := \bar{a}$, $k_{n+1} := k_{\bar{a}}$, $t_{n+1} := t_{\bar{a}}$, and $\tau_{n+1} = \tau_n + t_{n+1} \mathbf{e}_{n+1}^T$. Thus $\tau_{n+1}(n+1) > \phi(a_n)$ and $C + \tau_{n+1} v_i = \phi(a_i)$, for each $1 \leq i \leq n+1$.

Set $\tau = \tau_k$. Since $[V]$ is reduced-spinal, $\tau f \neq \tau v_k$. Furthermore, $\tau f > \tau v_i$ since $tau(i+1) = t_{i+1} > \tau v_i$, for each $i = 0, 1, \ldots, k-1$. Therefore $\tau f \notin \tau V$. □

We deduce that Lemma 5.1.11 is not applicable to any pre-spinal equation:[7]

Corollary 5.3.4. If $[\mathrm{D}]$ is pre-spinal and $K > 1$ then $[\mathrm{D}], K \not\models (\star\star)$.

Proof. Let $[\mathrm{D}] = (\mathbf{1_n}, \mathrm{D}) \in \mathcal{S}$ be pre-spinal. Then there exists a reduced-spinal $[V] = (f, V) \in \mathcal{A}$ such that $[V] = [\boldsymbol{\sigma}\mathrm{D}]$ for some substitution $\boldsymbol{\sigma}$. Fix $K > 1$ and let ϕ_K be the map $n \mapsto K^n$. By Lemma 5.3.3, $C + \tau V \subseteq \phi_K[\mathbb{N}]$ but $\tau f \notin \tau V$, for some $\tau \in \mathbb{F}^T$ and

[7] In fact, Lemma 5.3.3 implies that the argumentation used in Lemma 5.1.11 will be ineffective for proving register-admissibility for any machine $\mathtt{B}_\phi$, which simulates the register contents for a machine $\mathtt{B}$ by an injective function, without having more information about $\mathrm{Acc}(\mathtt{B})$.

$C \in \mathbb{N}$. Setting $\sigma = \tau\boldsymbol{\sigma} \in \mathbb{F}^T$, we obtain $C + \sigma\mathrm{D} \subseteq \phi_K[\mathbb{N}]$ but $\sigma\mathbf{1_n} \notin \sigma\mathrm{D}$. Setting $\sigma' = \sigma$ and $C' = C$ we deduce $[\mathrm{D}], K \not\models (\star\star)$. □

5.3.3 Pre-spinality. Let $\boldsymbol{\sigma}$ be a substitution and $\mathrm{D} \subseteq \mathbb{F}$. If $\boldsymbol{\sigma}\mathrm{D} = \{f\}$ for some $f \in \mathbb{F}$, we say $\boldsymbol{\sigma}$ *solves* D, and we write $\boldsymbol{\sigma}\mathrm{D} = f$. Similarly, if $\sigma \in \mathbb{F}^T$ such that $\sigma\mathrm{D} = \{a\}$ for some $a \in \mathbb{N}$, we say σ is a *solution* for D and write $\sigma\mathrm{D} = a$, and define $\mathrm{Sol}(\mathrm{D}) \subseteq \mathbb{F}^T$ to be the set of all solutions of D. Clearly, $\boldsymbol{\sigma}$ solves D iff $\boldsymbol{\sigma} = (\sigma_i)_{i=1}^k$ and σ_i is a solution for D for each $i = 1, \ldots, k$. Given non-empty $T \subseteq S \subseteq \mathrm{supp}(\mathrm{D})$, we say $\sigma \in \mathbb{F}^T$ is a (T, S)-solution for D if $\sigma \in \mathrm{Sol}(\mathrm{D})$ and σ is (T, S)-positive.

Let $\boldsymbol{\sigma} = (\sigma_i)_{i=1}^k$ be a substitution for $k \geq 1$. Suppose D is a $\boldsymbol{\sigma}$-spine witnessed by $\boldsymbol{\sigma}\mathrm{D} = V$, for some spine V. Viewed as a matrix equation and rearranging the columns, this substitution naturally partitions the columns of D so that $\boldsymbol{\sigma}\mathrm{D} = [\boldsymbol{\sigma}\mathrm{D}_0^{\boldsymbol{\sigma}} \cdots \boldsymbol{\sigma}\mathrm{D}_k^{\boldsymbol{\sigma}}] = [v_0 \cdots v_k]$, i.e., $\mathrm{D}_j^{\boldsymbol{\sigma}} := \{d \in \mathrm{D} : \boldsymbol{\sigma}d = v_j\}$ and so $\boldsymbol{\sigma}$ solves D_j. In other words, we use the flexibility of moving columns in aims to display a better presentation.

For each $0 \leq j \leq k$, we define $S_j \subseteq \mathbf{n}$, for $j \geq 1$ with $i \in S_j$ iff $\mathrm{D}_j^{\boldsymbol{\sigma}}[i] \neq \mathbf{0}$ and $\mathrm{D}_l^{\boldsymbol{\sigma}}[i] = 0$ for all $0 \leq l < j$, and $S_0 = \mathbf{n} \setminus S_1^{\uparrow}$, where $S_j^{\uparrow} := \bigcup_{j>i} S_i$. Now, $\sigma_i\mathrm{D}_j^{\boldsymbol{\sigma}} = v_j(i)$ and since $i \in \mathrm{supp}(v_i) \subseteq \{1, \ldots, i\}$, $\sigma_i\mathrm{D}_i^{\boldsymbol{\sigma}} > 0$ and $\sigma_i\mathrm{D}_j^{\boldsymbol{\sigma}} = 0$ only if $i > j$, for each $i > 1$. Therefore $S_j \neq \emptyset$ for each $j \geq 1$ and so $\mathfrak{b}_{\boldsymbol{\sigma}} := (S_0, \ldots, S_k)$ partitions $\mathbf{n}$, with S_0 possibly empty. Furthermore, every row of $\mathrm{D}_i^{\boldsymbol{\sigma}}[S_i]$ is non-zero, while $\mathrm{D}_j^{\boldsymbol{\sigma}}[S_i] = \mathbf{0}$ when $i > j$, for each $i \geq 1$. For the same reason, $\sigma_i[S_i] \neq \mathbf{0}$ but $\sigma_i[S_j] = \mathbf{0}$ for each $j < i$, i.e., σ_i is $(S_i, S_i^{\uparrow})$-positive.

For each $k \geq j \geq 1$ define $\mathrm{D}_{*j}^{\mathfrak{b}\boldsymbol{\sigma}} := \{d \in \mathrm{D} : \mathrm{supp}(d) \cap S_j \neq \emptyset\} \setminus \bigcup_{i>j} \mathrm{D}_{*i}^{\mathfrak{b}\boldsymbol{\sigma}}$ and $\mathrm{D}_{*0}^{\mathfrak{b}\boldsymbol{\sigma}} = \mathrm{D} \setminus \bigcup_{i=1}^k \mathrm{D}_{*i}^{\mathfrak{b}\boldsymbol{\sigma}}$. Then $\mathrm{D}_{*j}^{\mathfrak{b}\boldsymbol{\sigma}} = \mathrm{D}_j^{\boldsymbol{\sigma}}$ for each $j \geq 0$. So $\mathrm{D}^{\boldsymbol{\sigma}} = \mathrm{D}^{\mathfrak{b}\boldsymbol{\sigma}} := [\mathrm{D}_{*0}^{\mathfrak{b}\boldsymbol{\sigma}} \cdots \mathrm{D}_{*k}^{\mathfrak{b}\boldsymbol{\sigma}}]$. Visually, we rearrange D and $\boldsymbol{\sigma}$ into *upper-triangle block matrices* $\mathrm{D}^{\mathfrak{b}\boldsymbol{\sigma}}$ and $\boldsymbol{\sigma}^{\mathfrak{b}\boldsymbol{\sigma}}$ so that:

$$\begin{bmatrix} \mathbf{0} & \boldsymbol{\sigma}_{11}^{\mathfrak{b}_{\boldsymbol{\sigma}}} & \cdots & \boldsymbol{\sigma}_{1i}^{\mathfrak{b}_{\boldsymbol{\sigma}}} & \cdots & \boldsymbol{\sigma}_{1k} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\sigma}_{ii}^{\mathfrak{b}_{\boldsymbol{\sigma}}} & \cdots & \boldsymbol{\sigma}_{ik}^{\mathfrak{b}_{\boldsymbol{\sigma}}} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \boldsymbol{\sigma}_{kk}^{\mathfrak{b}_{\boldsymbol{\sigma}}} \end{bmatrix} \begin{bmatrix} \mathrm{D}_{00}^{\mathfrak{b}_{\boldsymbol{\sigma}}} & \mathrm{D}_{01}^{\mathfrak{b}_{\boldsymbol{\sigma}}} & \cdots & \mathrm{D}_{0i}^{\mathfrak{b}_{\boldsymbol{\sigma}}} & \cdots & \mathrm{D}_{0k}^{\mathfrak{b}_{\boldsymbol{\sigma}}} \\ \mathbf{0} & \mathrm{D}_{11}^{\mathfrak{b}_{\boldsymbol{\sigma}}} & \cdots & \mathrm{D}_{1i}^{\mathfrak{b}_{\boldsymbol{\sigma}}} & \cdots & \mathrm{D}_{1k}^{\mathfrak{b}_{\boldsymbol{\sigma}}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathrm{D}_{ii}^{\mathfrak{b}_{\boldsymbol{\sigma}}} & \cdots & \mathrm{D}_{ik}^{\mathfrak{b}_{\boldsymbol{\sigma}}} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathrm{D}_{kk}^{\mathfrak{b}_{\boldsymbol{\sigma}}} \end{bmatrix} = V,$$

Figure 5.3: Spines as products of upper-triangular block matrices

where $\mathrm{D}_{ij}^{\mathfrak{b}_{\boldsymbol{\sigma}}} := \mathrm{D}_{*j}^{\mathfrak{b}_{\boldsymbol{\sigma}}}[S_i]$ and $\boldsymbol{\sigma}_{ij}^{\mathfrak{b}_{\boldsymbol{\sigma}}} := \sigma_i[S_j]$. That is, starting from the right, we collect all rows $\mathrm{D}[i]$, that are only non-zero in $\mathrm{D}_k^{\boldsymbol{\sigma}}$, into a collection $\mathrm{D}_{k*}^{\mathfrak{b}_{\boldsymbol{\sigma}}}$ and move it to the bottom. We then repeat this process for the submatrix of D with the rows $\mathrm{D}_{j*}^{\mathfrak{b}_{\boldsymbol{\sigma}}}$ and columns D_j removed, for $j \leq k$.

Let $\mathrm{R} \subseteq \mathbb{F}$ with $\mathrm{supp}(R) = \mathbf{n}$, and let $\mathfrak{b} = (X_0, \ldots, X_k)$ a tuple of subsets $X_0, \ldots, X_n$ that partition $\mathbf{n}$, where we allow X_0 possibly empty. Define $\mathrm{R}^{\mathfrak{b}} := [\mathrm{R}_{*0}^{\mathfrak{b}} \ \cdots \ \mathrm{R}_{*k}^{\mathfrak{b}}]$, where

$$\mathrm{R}_j^{\mathfrak{b}} := \{r \in \mathrm{R} : \mathrm{supp}(\mathrm{R}) \cap X_j \neq \emptyset\} \setminus \bigcup_{i>j} \mathrm{R}_{*i}^{\mathfrak{b}}$$

for each $0 < j \leq k$, and $\mathrm{R}_{*0}^{\mathfrak{b}} = \mathrm{R} \setminus \bigcup_{j=1}^{k} \mathrm{R}_{*j}^{\mathfrak{b}}$. We say $\mathfrak{b}$ is a *blocking for* R iff the sets $\mathrm{R}_{*j}^{\mathfrak{b}}$ are nonempty for each $0 < j \leq k$. Given $\mathfrak{b} = (X_0, \ldots, X_k)$, we define $X_i^{\uparrow} := \bigcup_{j=i}^{k} X_i$. Note that there are only finitely many possible blockings for R, and if $(X_0, \ldots, X_k)$ is a blocking then $(\mathbf{n} \setminus X_i^{\uparrow}, X_i, \ldots, X_k)$ is a blocking for each $i = 1, \ldots, k$.

From the observation above, if D is a $\boldsymbol{\sigma}$-spine then $\mathfrak{b}_{\boldsymbol{\sigma}}$ is a blocking for D. Moreover, if $\boldsymbol{\sigma} = (\sigma_i)_{i=1}^{k}$, then σ_i is a $(S_i, S_i^{\uparrow})$-solution for each $\mathrm{D}_{*j}^{\mathfrak{b}_{\boldsymbol{\sigma}}}$. On the other hand, if $\mathfrak{b} = (S_0, \ldots, S_k)$ is a blocking for D and there exists $\sigma_1, \ldots, \sigma_k \in \bigcap_{j=1}^{k} \mathrm{Sol}(\mathrm{D}_{*j}^{\mathfrak{b}})$ such that each σ_i is $(S_i, S_i^{\uparrow})$-positive, then D is a $\boldsymbol{\sigma}$-spine for $\boldsymbol{\sigma} := (\sigma_i)_{i=1}^{k}$. If $[\mathrm{D}] = (\mathbf{1}_{\mathbf{n}}, \mathrm{D})$, then $[\boldsymbol{\sigma}\mathrm{D}]$ is a spine if $\sigma_j \mathbf{1}_{\mathbf{n}} \neq \sigma_j \mathrm{D}_{*k}^{\mathfrak{b}}$ for some $j \geq 1$. Since we need only consider reduced-spines, we conclude:

Theorem 5.3.5. Let $[\mathrm{D}] = (\mathbf{1_n}, \mathrm{D}) \in \mathcal{S}$. Then $[\mathrm{D}]$ is pre-spinal if and only if $[\mathrm{D}]$ is integral or there exists a blocking $\mathfrak{b} = (S_0, \ldots, S_k)$ of D, with $k \geq 1$, and $\sigma_1, \ldots, \sigma_k \in \bigcap_{j=1}^{k} \mathrm{Sol}(\mathrm{D}^{\mathfrak{b}}_{*j})$, where each σ_i is $(S_i, S_i^{\uparrow})$-positive, but $\sigma_1 \mathbf{1_n} \neq \sigma_1 \mathrm{D}^{\mathfrak{b}}_{*k}$.

Example 5.3.1. Consider the simple equation

$$[\mathrm{R}] : wxyz \leq 1 \vee w \vee w^4x^2y \vee w^3y^2z \vee w^2xz^2.$$

Indexing the variables alphabetically, $[\mathrm{R}]$ is equivalent to $(\mathbf{1_4}, \mathrm{R}) \in \mathcal{S}$ where

$$\mathrm{R} = \{\mathbf{0}, (1,0,0,0), (4,2,1,0), (3,0,2,1), (2,1,0,2)\},$$

its natural presentation as a subset of $\mathbb{N}^4$. Observe that

$$\boldsymbol{\sigma}\mathrm{R}^{\mathfrak{b}} = \left[\begin{array}{c|ccc} 1 & 0 & 0 & 1 \\ \hline 0 & 1 & 1 & 1 \end{array}\right] \cdot \left[\begin{array}{c|cccc} 0 & 1 & 4 & 3 & 2 \\ \hline 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 & 2 \end{array}\right] = \left[\begin{array}{c|cccc} 0 & 1 & 4 & 4 & 4 \\ \hline 0 & 0 & 3 & 3 & 3 \end{array}\right] = V,$$

where $\mathfrak{b} = (\emptyset, \{1\}, \{2,3,4\})$ and $\boldsymbol{\sigma} = (\sigma_i)_{i=1}^2$, where $\sigma_1 = (1,0,0,1)^T; \sigma_2 = (0,1,1,1)^T$. Since $\boldsymbol{\sigma}\mathbf{1_4} = (2,3) \notin \boldsymbol{\sigma}\mathrm{R}$ and V is a spine, $(\boldsymbol{\sigma}\mathbf{1_4}, \boldsymbol{\sigma}\mathrm{R})$ is spinal. Therefore $[\mathrm{R}]$ is pre-spinal. Reverting to the multiplicative notation, this substitution shows

$$\mathsf{CRL} + [\mathrm{R}] \models x^2y^3 \leq 1 \vee x \vee x^4y^3.$$

5.3.4 Solutions in $\mathbb{R}^n$. Let $v \in \mathbb{R}^n$ and $M \subseteq \mathbb{R}^n$. We say a vector v is *orthogonal* to the set M if $v^T M = 0$. We say $v \in \mathbb{R}^n$ is *strictly (strongly) positive* if $v \neq \mathbf{0}$ and $v(i) \geq 0$ ($v(i) > 0$) for each $i \in \mathbf{n}$. The set X^n_+ (X^n_{++}) denotes the set of all strictly

(strongly) positive vectors in X^n, called the *strictly (strongly) positive orthant* in X^n, where $X \in \{\mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$. Note that $\mathrm{span}(M)[S] = \mathrm{span}(M[S])$ for any $M \subseteq \mathbb{R}^n$ and $S \subseteq \mathbf{n}$.

Let $\sigma \in \mathbb{F}^T$, $\mathrm{D} \subseteq \mathbb{F}$ and $S \subseteq \mathbf{n}$. Then σ is a solution for $\mathrm{D}[S]$ iff $\sigma[S]$ is orthogonal to $\bar{\mathrm{D}}[S]$ in $\mathbb{R}^n$, where $\bar{\mathrm{D}} := \{d - \bar{d} : d \in \mathrm{D}\}$ for any fixed $\bar{d} \in \mathrm{D}$. Hence, if $T \subseteq S$, then there exists a (T, S)-solution for D iff there exists a T-solution for $\mathrm{D}[S]$ iff there exists a T-positive $v \in \mathbb{R}^n_+$ orthogonal to $\bar{\mathrm{D}}[S]$.[8] We recall a theorem of the alternatives for positive solutions to linear systems.

Theorem 5.3.6 ([21])**.** Let $M \subseteq \mathbb{R}^n$ be a set of vectors. Then exactly one of the following holds:

1. There exists a strictly (strongly) positive $v \in \mathbb{R}^n$ orthogonal to M, or
2. $\mathrm{span}(M)$ intersects the strongly (strictly) positive orthant of $\mathbb{R}^n$.

Corollary 5.3.7. Let $M \subseteq \mathbb{R}^n$ and $S \subseteq \{1, \ldots, n\}$ be nonempty. If there is no S-positive vector $v \in \mathbb{R}^n_+$ orthogonal to M then there exists a strictly positive $w \in \mathrm{span}(M)$ with $S \subseteq \mathrm{supp}(w)$.

Proof. The assumption implies, in particular, there is no strongly positive vector orthogonal to M. By Theorem 5.3.6, there exists a strictly positive $v \in \mathrm{span}(M)$. Proceeding inductively, if $n = 1$ then $\mathrm{supp}(v) = \{1\} = S$ and we are done. Suppose the claim is true for all $1 \leq m < n$. Let $Y = \{i : \exists u \in \mathrm{span}(M) \cap \mathbb{R}^n_+ \text{ with } u(i) > 0\}$ and $X := \mathbf{n} \setminus Y$. Since v is strictly positive, $Y \neq \emptyset$ and there exists $w_1 \in \mathrm{span}(M)$ with $Y = \mathrm{supp}(w_1)$. If $S \subseteq Y$ then we are done. Otherwise, $T := X \cap S \neq \emptyset$ and we consider the projection $M[X] \subseteq \mathbb{R}^{|X|}$. Since a T-positive $u \in \mathbb{R}^{|X|}_+$ orthogonal to $M[X]$ would also serve as an S-positive vector in $\mathbb{R}^n_+$ orthogonal to M, there must be no T-positive $u \in \mathbb{R}^n_+$ orthogonal to $M[X]$. Since $1 \leq |X| < n$, by the induction hypothesis we have that there exists

[8] For the reverse direction, since $\bar{\mathrm{D}}[S] \subseteq \mathbb{Z}^n$, by Gaussian Elimination we may assume $v \in \mathbb{Q}^n_+$, and so $\sigma = t \cdot v \in \mathbb{Z}^n_+$ for some $t \in \mathbb{N}$.

$w' \in \mathrm{span}(M[X])$ with $T \subseteq \mathrm{supp}(w')$. Let $w_2 \in \mathrm{span}(M)$ such that $w_2[X] = w'$. Let $t = \max\{|w_2(i)| : i \in Y\}$. Then since $w_1[X] = \mathbf{0}$ and $w_2[X]$ is strictly positive, we have that $w := (t+1)w_1 + w_2 \in \mathrm{span}(M) \cap \mathbb{R}^n_+$ such that $S \subseteq \mathrm{supp}(w)$. □

Corollary 5.3.8. Let $M \subseteq \mathbb{R}^n$ and $T \subseteq S \subseteq \mathbf{n}$ be non-empty. If there is no T-positive $v \in \mathbb{R}^n_+$ orthogonal to $M[S]$ then there exists $L \in \mathbb{N}$ such that, for any $v \in \mathbb{R}^n_+$ orthogonal to M, $v(i) \leq L \cdot \max\{v(j) : j \in \mathbf{n} \setminus S\}$ for each $i \in T$.

Proof. By Corollary 5.3.7, there exists a strictly positive vector $\bar{w} \in \mathrm{span}(M[S])$ with $T \subseteq \mathrm{supp}(\bar{w})$. So there is $w \in \mathrm{span}(M)$ with $w[S] = \bar{w}$. Let $X = \mathbf{n} \setminus S$ and $L = |X| \cdot \max\{|w(j)| : j \in X\}$. Suppose $v \in \mathbb{R}^n_+$ is orthogonal to M, then $v^T w = 0$ and, since $T \subseteq \mathrm{supp}(w)$ we have

$$v(i) \leq \sum_{j \in S} w(j)v(j) = \sum_{j \in X} -w(j)v(j) \leq \sum_{j \in X} |w(j)|v(j) \leq L \cdot \max\{v(j) : j \in X\},$$

for each $i \in T$. □

5.3.5 ($\star$) and ($\star\star$). Recall $\Delta[\mathrm{D}] := \sum_{i=1}^n \max\{|d(i) - d'(i)| : d, d' \in \mathrm{D}\}$ for finite $\mathrm{D} \subseteq \mathbb{F}$. For each $K > 1$, let ϕ_K be the mapping $n \mapsto K^n$.

Lemma 5.3.9. Let $d, d' \in \mathbb{F}$ and $K > 1$ and suppose $C + \sigma d, C + \sigma d' \in \phi_k[\mathbb{N}]$ with $\sigma d > \sigma d'$, for some $C \in \mathbb{N}$ and $\sigma \in \mathbb{F}^T$. If there exists $L \in \mathbb{N}$ such that $\sigma(i) \leq L \cdot (\sigma d')$ for each $i \in \mathrm{supp}\{d, d'\}$, then $K \leq L \cdot \Delta[d, d'] + 1$.

Proof. Suppose $C + \sigma \mathrm{D} = K^{a+b}$ and $C + \sigma d' = K^a$ are distinct, for some $a \geq 0$ and $b \geq 1$. On the one hand, $\sigma d - \sigma d' = K^a(K^b - 1) \geq K^a(K-1)$. While on the other hand, $\sigma d - \sigma d' \leq \sigma|d - d'| \leq LK^a\Delta[d, d']$. Hence $K \leq L\Delta[d, d'] + 1$. □

For $\sigma \in \mathbb{F}^T$, define $\mathrm{D}^\sigma = [\mathrm{D}^\sigma_0 \ \cdots \ \mathrm{D}^\sigma_k]$, where $\sigma(\mathrm{D} \cup \{\mathbf{0}\}) = \{n_0, \ldots, n_k\}$, for $0 = n_0 < \cdots < n_k$, and $\mathrm{D}^\sigma_j := \{d \in \mathrm{D} : \sigma d = n_j\}$. We say *$\sigma$ defines a blocking for* D if $\mathrm{D}^\mathfrak{b} = \mathrm{D}^\sigma$ for some blocking $\mathfrak{b}$ for D, i.e., $\sigma \in \bigcap_{j=0}^k \mathrm{Sol}(\mathrm{D}^\mathfrak{b}_{*i})$ with $\sigma\mathrm{D}^\mathfrak{b}_{*k} > \cdots > \sigma\mathrm{D}^\mathfrak{b}_{*0} = 0$.

Lemma 5.3.10. Let $\mathrm{D} \subseteq \mathbb{F}$ be finite and $K > 1$. Suppose $\sigma \in \mathbb{F}^T$ and $C \in \mathbb{N}$ such that $C + \sigma\mathrm{D} \subseteq \phi_K[\mathbb{N}]$. If $K > \Delta[\mathrm{D}] + 1$ then σ defines a blocking for D.

Proof. Suppose $\mathrm{supp}(\mathrm{D}) = \mathbf{n}$. If $k = 0$ then $\mathfrak{b} = (\mathbf{n})$ is the blocking, so consider $k \geq 1$. We claim for each $d \in \mathrm{D}_k^\sigma$, $\mathrm{supp}(d) \setminus \mathrm{supp}(\bigcup_{i=0}^{k-1} \mathrm{D}_i^\sigma) \neq \emptyset$. Supposing otherwise would entail that for each $i \in \mathrm{supp}(d)$ there exists $j < k$ and $d' \in \mathrm{D}_j^\sigma$ such that $i \in \mathrm{supp}(d')$, implying $K \leq \Delta[\mathrm{D}] + 1$ by Lemma 5.3.9 and contradicting our assumption. Then $S_k := \{i \in \mathbf{n} : \mathrm{D}_l^\sigma[i] \neq 0 \Rightarrow l = k\}$ is nonempty and $\mathrm{D}_k^\sigma = \{d \in \mathrm{D} : \mathrm{supp}(d) \cap S_k \neq \emptyset\}$. Continuing in this way for $1 \leq j < k$, since $\mathrm{D}' \subseteq \mathrm{D}$ implies $\Delta[\mathrm{D}'] \leq \Delta[\mathrm{D}]$, the same argument shows $\mathrm{supp}(d) \setminus \mathrm{supp}(\bigcup_{i=0}^{j-1} \mathrm{D}_i^\sigma)$ is nonempty for each $d \in \mathrm{D}_j^\sigma$, and so $S_j := \{i \in \mathbf{n} : l \leq j \;\&\; \mathrm{D}_l^\sigma[i] \neq \mathbf{0} \Rightarrow l = j\}$ is nonempty with $\mathrm{D}_j^\sigma = \{d \in \mathrm{D} : \mathrm{supp}(d) \cap S_j \neq \emptyset\} \setminus \bigcup_{i=j+1}^{k} \mathrm{D}_j^\sigma$. By defining $S_0 = \mathrm{supp}(\mathrm{D}) \setminus S_1^\uparrow$, we conclude $\mathfrak{b} = (S_0, \ldots, S_k)$ is a blocking of D such that $\mathrm{D}^\mathfrak{b} = \mathrm{D}^\sigma$. □

Lemma 5.3.11. Let $\mathfrak{b} = (S_0, \ldots, S_k)$ be a blocking for D with $k \geq 1$. Suppose for some $i \geq 1$ there is no $(S_i, S_i^\uparrow)$-solution in $\bigcap_{j=0}^{k} \mathrm{Sol}(\mathrm{D}_{*j}^\mathfrak{b})$. Then there exists $L \in \mathbb{N}$ such that for any $\sigma \in \mathbb{F}^T$ with $\mathrm{D}^\sigma = \mathrm{D}^\mathfrak{b}$, if $C + \sigma\mathrm{D} \subseteq \phi_K[\mathbb{N}]$ then $K \leq L\Delta[\mathrm{D}] + 1$.

Proof. If $i = 1$, then $\sigma \in \mathrm{Sol}(\mathrm{D}_{*1}^\mathfrak{b})$ implies $\sigma\mathrm{D}_{*1}^\mathfrak{b} = 0$ by assumption, so $\mathrm{D}^\sigma \neq \mathrm{D}^\mathfrak{b}$ and the claim is vacuously satisfied. Suppose $i > 1$. Observe there exists a $(S_i, S_i^\uparrow)$-solution in $\bigcap_{j=0}^{k} \mathrm{Sol}(\mathrm{D}_{*j}^\mathfrak{b})$ iff there exists an S_i-positive $v \in \mathbb{R}^+$ orthogonal to $\bar{\mathrm{D}}^\mathfrak{b}[S_i^\uparrow]$. Since $\mathrm{D}^\sigma = \mathrm{D}^\mathfrak{b}$ implies $\sigma\mathrm{D}_{*j}^\mathfrak{b} > \sigma\mathrm{D}_{*j-1}^\mathfrak{b}$, by Corollary 5.3.8 and Lemma 5.3.9 the result follows. □

For $K > 1$ and $[\mathrm{D}] = (\mathbf{1_n}, \mathrm{D}) \in \mathcal{S}$, we write $[\mathrm{D}], K \models (\star)$ if and only if

$$\begin{gathered} \text{For all } \sigma \in \mathbb{F}^T \text{ and for all } C \in \mathbb{N}, \\ \text{if } C + \sigma d \text{ is a power of } K \text{ for each } d \in \mathrm{D} \\ \text{then there exists } \bar{d} \in \mathrm{D} \text{ such that } \sigma\bar{d} = \sigma\mathbf{1_n}; \end{gathered} \tag{$\star$}$$

i.e., $C + \sigma\mathrm{D} \subseteq \phi_K[\mathbb{N}]$ implies $\sigma\mathbf{1_n} \in \sigma\mathrm{D}$.

Lemma 5.3.12. If $[\mathrm{D}] \in \mathcal{U}$ then $[\mathrm{D}], K \models (\star)$ for all K sufficiently large.

Proof. We proceed by contraposition. Suppose $A := \{K \in \mathbb{N} : [\mathrm{D}], K \not\models (\star)\}$ is infinite. For each blocking $\mathfrak{b}$ of D, define $A_{\mathfrak{b}} \subseteq A$ via $K \in A_{\mathfrak{b}}$ iff $K \in A$ witnessed by $\sigma \in \mathbb{F}^T$ such that $\mathrm{D}^\sigma = \mathrm{D}^{\mathfrak{b}}$. Since A is infinite and there are only finitely many blockings of D, there exists $\mathfrak{b}$ such that $A_{\mathfrak{b}}$ is infinite by Lemma 5.3.10. Fix a blocking $\mathfrak{b} = (S_0, \ldots, S_k)$ for D such that $A_{\mathfrak{b}}$ is infinite. Let σ_1 be a witness to the failure of $(\star)$ for some $K \in A_{\mathfrak{b}}$ such that $\mathrm{D}^{\sigma_1} = \mathrm{D}^{\mathfrak{b}}$. If $k = 0$ then $[\mathrm{D}]$ must be integral and we are done. If $k \geq 1$, then $\sigma_1 \in \bigcap_{j=0}^{k} \mathrm{Sol}(\mathrm{D}^{\mathfrak{b}}_{*j})$ is $(S_1, S_1^{\uparrow})$-solution such that $\sigma_1 \mathbf{1_n} \neq \sigma \mathrm{D}^{\mathfrak{b}}_{*k}$. Furthermore, since $A_{\mathfrak{b}}$ is infinite there must be a $(S_i, S_i^{\uparrow})$-solution $\sigma_i \in \bigcap_{j=0}^{k} \mathrm{Sol}(\mathrm{D}^{\mathfrak{b}}_{*j})$ by Lemma 5.3.11, for each $1 \leq i \leq k$. Therefore, by Theorem 5.3.5, $[\mathrm{D}] \notin \mathcal{U}$. □

Theorem 5.3.13. Let $[\mathrm{D}] \in \mathcal{S}$. Then $[\mathrm{D}] \in \mathcal{U}$ if and only if there exists $N \in \mathbb{N}$ such that $[\mathrm{D}], K \models (\star\star)$ for every $K \geq N$.

Proof. The reverse direction follows from Lemma 5.3.4. We proceed by contradiction for the forward direction. Suppose $[\mathrm{D}] = (\mathbf{1_n}, \mathrm{D}) \in \mathcal{U}$ but for every $M \in \mathbb{N}$ there exists $K_M > M$ such that $[\mathrm{D}], K_M \not\models (\star\star)$, i.e., there exists $\sigma, \sigma' \in \mathbb{F}$ and $C, C' \in \mathbb{N}$ such that $C + \sigma\mathrm{D}, C' + \sigma'\mathrm{D} \subseteq \phi_{K_M}[\mathbb{N}]$ but for all $d \in \mathrm{D}$, either $\sigma \mathbf{1_n} \neq \sigma d$ or $\sigma' \mathbf{1_n} \neq \sigma' d$. By Lemma 5.3.10, σ and σ' each define a blocking for D if $K_M > \Delta[\mathrm{D}] + 1$. Since there are only finitely many blockings of D, there must exist a pair $\mathfrak{b}, \mathfrak{c}$ that witness this failure for every K_M in some infinite set $A \subseteq \mathbb{N}$. Let $\mathfrak{b} = (S_0, \ldots, S_k)$ and $\mathfrak{c} = (T_0, \ldots, T_l)$. Since $[\mathrm{D}] \in \mathcal{U}$, $[\mathrm{D}]$ is not integral and so $k, l \geq 1$.

Since A is infinite, there is a (S_k, S_k)-solution $\sigma_{\mathfrak{b}} \in \mathbb{F}^T$ for $\mathrm{D}^{\mathfrak{b}}_{*k}$ by Lemma 5.3.11. Thus D is a $(\sigma_{\mathfrak{b}})$-spine. Now, $\sigma_{\mathfrak{b}} \mathbf{1_n} = \sigma_{\mathfrak{b}} \mathrm{D}^{\mathfrak{b}}_k$ since otherwise $[\mathrm{D}]$ would be pre-spinal, contradicting $[\mathrm{D}] \in \mathcal{U}$. So let $t_{\mathfrak{b}} := \sigma_{\mathfrak{b}} \mathbf{1_n} = \sigma_{\mathfrak{b}} \mathrm{D}_K$, and note $t_{\mathfrak{b}} > 0$ since $k \geq 1$. By symmetry, there is a (T_l, T_l)-solution $\sigma_{\mathfrak{c}}$ for $\mathrm{D}^{\mathfrak{c}}_{*l}$ such that $t_{\mathfrak{c}} := \sigma_{\mathfrak{c}} \mathbf{1_n} = \sigma_{\mathfrak{c}} \mathrm{D}^{\mathfrak{c}}_l > 0$.

We claim that S_k and T_l are disjoint. Since $[\mathrm{D}] \in \mathcal{U}$, by Lemma 5.3.12 there exists $N \in \mathbb{N}$ such that $[\mathrm{D}], K \models (\star)$ for every $K \geq N$. Let $K \in A$ with $K > \max\{N, \Delta[\mathrm{D}]+1\}$, and σ, σ' falsifying $(\star\star)$ for some with $\mathrm{D}^{\mathfrak{b}} = \mathrm{D}^{\sigma}$ and $\mathrm{D}^{\mathfrak{c}} = \mathrm{D}^{\sigma'}$. Since $K > N$, $(\star)$ implies that $\sigma\mathbf{1_n} \in \sigma\mathrm{D}^{\mathfrak{b}}$ ($\sigma'\mathbf{1_n} \in \sigma\mathrm{D}^{\mathfrak{c}}$). In addition, $K > \Delta[\mathrm{D}] + 1$ further implies $\sigma\mathbf{1_n} = \sigma\mathrm{D}^{\mathfrak{b}}_{*k}$ ($\sigma'\mathbf{1_n} = \sigma'\mathrm{D}^{\mathfrak{c}}_{*l}$) by Lemma 5.3.9. Since σ, σ' falsify $(\star\star)$, there is no $d \in \mathrm{D}$ such that $\sigma d = \sigma\mathbf{1_n}$ and $\sigma' d = \sigma'\mathbf{1_n}$. So $\mathrm{D}^{\mathfrak{b}}_{*k} \cap \mathrm{D}^{\mathfrak{c}}_{*l} = \emptyset$, and hence $S_k \cap T_l = \emptyset$ by definition of $\mathfrak{b}, \mathfrak{c}$ being blockings. Hence $\sigma_{\mathfrak{b}}\mathrm{D}^{\mathfrak{c}}_{*l} = 0$ and $\sigma_{\mathfrak{c}}\mathrm{D}^{\mathfrak{b}}_{*k} = 0$. Let $X_1 := \mathrm{D}^{\mathfrak{b}}_{*k} \cup \mathrm{D}^{\mathfrak{c}}_{*l}$ and $X_0 := \mathrm{D} \setminus X_1$.

Hence, for $\bar{\sigma} := t_{\mathfrak{c}}\sigma_{\mathfrak{b}} + t_{\mathfrak{b}}\sigma_{\mathfrak{c}}$, it follows that $\bar{\sigma}X_1 = t_{\mathfrak{b}}t_{\mathfrak{c}} > \bar{\sigma}X_0 = 0$, but $\bar{\sigma}\mathbf{1_n} = 2t_{\mathfrak{b}}t_{\mathfrak{c}} > t_{\mathfrak{b}}t_{\mathfrak{c}}$. Therefore $[\mathrm{D}]$ is a $(\bar{\sigma})$-spine, contradicting $[\mathrm{D}] \in \mathcal{U}$. □

Corollary 5.3.14. If $\Sigma \subseteq \mathcal{U}$ is finite then there exists $K > 1$ such that $\Sigma, K \models (\star\star)$.

Proof. For each $[\mathrm{D}] \in \Sigma$ there exists N_{D} such that $[\mathrm{D}], K \models (\star\star)$ for all $K \geq N_{\mathrm{D}}$. Since Σ is finite, let $K := \max\{N_{\mathrm{D}} : [\mathrm{D}] \in \Sigma\}$. Then $\Sigma, K \models (\star\star)$ by definition. □

Chapter 6: Concluding remarks

We conclude this thesis with remarks about related results and a list of open problems.

6.1 The class $\mathcal{U}$ and known results

As our construction of algebraic machines in Chapter 4 was inspired by both [14] and [5], there are many connections between these manuscripts and the results of Chapter 5. We wish to briefly mention here the general scope of the results obtained in [14] and [5], natural generalizations of them, and their relation to the class $\mathcal{U}$. We note that the constructions in [14] and [5] properly require non-commutativity. Although we show there is overlap between the consequences of [14] and Section 5.2, we note that the our results for extensions of CRL are novel.

6.1.1 Horčík and the word problem for non-commutative varieties. In [14], Horčík proves that the word problem for $\mathsf{RL} + [\mathsf{k}_n^m]$ is undecidable for any knotted equation $[\mathsf{k}_n^m]$ for the values $n \neq m$ where $m \geq 2$ and $n \geq 1$. I.e., for all expansive knotted equations and all non-mingly compressive knotted equations. The argument used to establish this fact involves a residuated frames construction as in Section 4.1.1.

In particular, although not explicitly stated in [14], all equations present in the quasi-equations used for the encoding are of monoid-type in the fragment $\{\leq, \cdot, 1\}$. Hence, [14] in fact establishes that the $\{\leq, \cdot, 1\}$-fragment of the word problem for $\mathsf{RL} + [\mathsf{k}_n^m]$ in undecidable for the knotted equations described above.

We observe that since all such equations are knotted, they are by definition spinal and therefore are not members of $\mathcal{U}$. In fact, the result captures a broad class of pre-spinal equations for which our methods in Chapter 5 are unable to address. The residuated frame $\mathbf{W}$ that Horčík constructs is such that $\mathbf{W}^+ \models (x^3 \leq x^2) \& (x \leq x^2)$. By the same argument

in Theorem 4.1.5, Horčík shows that any subvariety $\mathcal{V} \subseteq \mathsf{RL}$ containing $\mathbf{W}^+$ will have an undecidable word problem. Consequently, if $\mathsf{RL} \models [(\forall x)(x^3 \leq x^2)\&(x \leq x^2)] \Rightarrow (\forall \overline{x})[\mathrm{R}]$ for some simple equation $[\mathrm{R}]$, then $\mathbf{W}^+ \in \mathsf{RL} + [\mathrm{R}]$. In particular (by Lem. 2.7 [14]):

Proposition 6.1.1 ([14])**.** Let $[\mathrm{R}] = (\mathbf{1_n}, \mathrm{R})$ be a nontrivial simple equation. If R contains a square, i.e., $ux^2v \in \mathrm{R}$ for some monoid terms u, v, x, then $\mathbf{W}^+ \in \mathsf{RL} + [\mathrm{R}]$. In particular, for any non-mingly single-variable equation $[\mathrm{R}]$, the word problem for $\mathsf{RL} + [\mathrm{R}]$ is undecidable.

Although this result is remarkably encompassing, there are members of $\mathcal{U}_{-1}$ for which [14] does not explicitly capture. We can use the very same function which Horčík utilized to ensure a language of square-free words (ensuring $\mathbf{W}^+ \models [(x^3 \leq x^2)\&(x \leq x^2)]$), to obtain a simple equation for which the above proposition is not applicable. Consider the alphabet $\Sigma = \{x, y, z\}$ and the free semigroup Σ^+ generated by Σ. Let $h : \Sigma^+ \to \Sigma^+$ be the semigroup homomorphism defined as follows:

$$\begin{aligned} h(x) &= xyz, \\ h(y) &= xz, \quad \cdot \\ h(z) &= y. \end{aligned}$$

As shown in [18], the n-th composition $h^n(x)$ is square-free, for any $n \geq 0$.

Consider the simple equation $[\mathrm{R}] : x \leq x^2 \vee x^3$. Now, $[\mathrm{R}] \in \mathcal{U}_{-1}$ and R contains a square. Both Theorem 5.2.6 and [14] entail that the $\{\leq, \cdot, 1\}$-fragment of the word problem for $\mathsf{RL} + [\mathrm{R}]$ is undecidable. Consider now the equation $[h\mathrm{R}] : h(x) \leq h^2(x) \vee h^3(x)$. By the definition of h, $[h\mathrm{R}]$ is given by

$$[h\mathrm{R}] : xyz \leq xyzxzy \vee xyzxzyxyzyxz,$$

and the right hand side is square-free. On the one hand, the argument in [14] seemingly fails for the equation $[h\mathrm{R}]$. On the other hand, $[h\mathrm{R}] \in \mathcal{U}_{-1}$ and therefore by Theorem 5.2.6, the $\{\leq, \cdot, 1\}$-fragment of the word problem for $\mathsf{RL} + [h\mathrm{R}]$ is undecidable.[1] In this way, infinitely many examples for which Theorem 5.2.6 hold but [14] seemingly fails can be constructed as above.

6.1.2 Chvalovský & Horčík and the non-commutative varieties. In [5], Chvalovský and Horčík prove that for every expansive knotted equation $[\mathsf{k}_n^m]$, i.e., for $m > n > 0$, $\mathsf{RL} + [\mathsf{k}_n^m]$ has an undecidable equational theory. In particular, they establish the remarkable fact that provability in $\mathbf{FL}_\mathsf{c}$ is undecidable. The main idea developed in [5] was to obtain a deduction theorem in which the undecidability of the word problem provided in [14] could be bootstrapped to the equational theory.

As in [14], the primary focus of [5] was to investigate expansive knotted equations, contraction in particular. However their result is general enough to establish that expansive equations, as defined in Section 2.4, admit the same property. That is, $\mathsf{RL} + [E]$ has an undecidable equational theory for any expansive equation $[E]$. As in [14], the challenge was to create an encoding that maintained the property that only square-free words are accepted to ensure instances of $[\mathsf{k}_n^m]$ are admissible. This is achieved by their so-called *Conditional String-Rewriting Systems* (CSRS). Now, expansive rules are of the form

$$[E] : x^n \leq \bigvee_{p \in P} x^p,$$

for some finite nonempty set $P \subseteq \mathbb{N}$ such that $p > n$ for each $p \in P$. As observed in Equation (4.12), if $[\mathrm{R}_E]$ is the equivalent simple equation for $[E]$, then $[\mathrm{R}_E]$ contains

[1] It is straightforward to verify $[h\mathrm{R}] \in \mathcal{U}_{-1}$. Indeed, $[h\mathrm{R}]_{\mathrm{com}} : xyz \leq x^2y^2z^2 \vee x^3y^3z^3$. Since the corresponding set of vectors is $\overline{h\mathrm{R}} := \{(2,2,2),(3,3,3)\}$, there is no non-zero $\sigma \in \mathbb{N}^3$ such that $|\sigma\overline{h\mathrm{R}}| = 1$ and $\sigma\overline{\mathbf{1_3}} \notin \sigma\overline{h\mathrm{R}}$. Hence we obtain $[h\mathrm{R}] \in \mathcal{U}$. Since no variable appears precisely once in each joinand in $h\mathrm{R}$, $h\mathrm{R} \in \mathcal{U}_{-1}$.

a square, i.e., $ux^2v \in \mathrm{R}_e$ for some monoid terms u, v, x. As a result (Thm. 3.5 [5]), their CSRS language L is trivially closed under the equation $[\mathrm{R}_E]$ since it L only accepts square-free words. That is, $[\mathrm{R}_E]$ is admissible in the language L.

The last step in [5] was to ensure the completeness of the encoding, i.e., L accepts some word if and only if some specific equation is satisfied in $\mathsf{RL} + [\mathsf{k}_n^m]$. This is achieved, on the one hand, by a cleverly constructed formula (§4 [5]) using so-called *atomic* CSRSs. On the other hand, the only role that the expansive knotted equation plays in the deduction theorem if for carrying out the instructions of the atomic CSRS (Lem. 4.1 [5]). That is, merely utilizing the fact that $\mathsf{RL} + [\mathsf{k}_n^m]$ is negatively n-potent (see Section 2.5).

It is clear then that since an expansive equation $[E]$ is such that (i) $[\mathrm{R}_E]$ contains squares, and (ii) the variety $\mathsf{RL} + [\mathrm{R}_E]$ is negatively n-potent, the variety $\mathsf{RL} + [\mathrm{R}_E]$ has an undecidable equational theory (Thm. 3.5, Thm. 4.4, §5.2, §5.3 [5]).

6.2 Open problems and future work

Lastly, we conclude with a list of open problems that arise from the contents of this thesis.

1. Let be a set of simple equations Σ and $[A] = (a_0, A)$ a proper ISR-equation. By Lemma 2.3.3, $\mathsf{RL} + \Sigma \models [A]$ iff $\mathsf{ISR} + \Sigma \models [A]$ iff $A \vdash_\Sigma a_0$. We observed in Section 3.2, that determining whether or not $A \vdash_\Sigma a_0$ is recursively enumerable. We gave sufficient condition for decidability in Theorem 3.2.1 and Theorem 3.2.2. Is it decidable in general? I.e., is the equational theory of $\mathsf{ISR} + \Sigma$ always decidable, or does there exists a special set of simple equations Σ such that $\mathsf{ISR} + \Sigma$ has an undecidable equational theory?

2. In Section 3.4, we show that the decision procedure for potent varieties of CRL extended by finitely many simple rules not only primitive recursive, but at worst doubly-exponential. Can this upper bound for complexity be lowered to, say, an exponential bound?

3. Let $m > n > 0$ and $[\mathsf{k}_n^m]$ be an expansive knotted equation. By [23] and Section 4.4, there is no primitive recursive decision procedure for the $\{\vee, \cdot, 1\}$-fragment of the quasi-equational theory for $\mathsf{CRL} + [\mathsf{k}_n^m]$. What can be said about the complexity of the word problem for this fragment? Specifically, is the word problem for $\{\vee, \cdot, 1\}$-fragment of $\mathsf{CRL} + [\mathsf{c}]$ primitive recursive?

4. Continuing from the above, since these varieties are commutative [23] and Section 4.4 relied on the presence of $\vee$ to simulate zero-test instructions for ACMs. What is the complexity of the quasi-equational theory (or even the word problem) for the $\{\leq, \mathrm{cot}, 1\}$-fragment of these varieties?

5. For the compressive knotted equation $[\mathsf{k}_m^n]$ with $m > n > 0$, $\mathsf{CRL} + [\mathsf{k}_m^n]$ has the FEP. Does $\mathsf{CRL} + [\mathsf{k}_m^n]$ admit a primitive recursive decision procedure? More specifically, what is a complexity lower bound for $\mathsf{CRL} + (x^3 \leq x^2)$? Can the construction in [23] show there is no primitive recursive decision procedure for $\mathsf{CRL} + (x^3 \leq x^2)$?

6. In [14], it is established that $\mathsf{RL} + [\mathsf{k}_n^m]$ has an undecidable word problem for any $n \geq 1$ and $m \geq 2$. In fact, this result holds for any single variable equation $x^n \leq \bigvee_{p \in P} x^p$, for $n \geq 1$, so long as the set $P \not\subseteq \{0, 1\}$. I.e., [14] does not cover equations of the form $x^n \leq x$, $x^n \leq x \vee 1$, and $x^n \leq 1$ (which is equivalent $x \leq 1$ in RL).[2] It has been known extensions by $x \leq 1$ and $x^2 \leq x$ have decidable universal theories (and hence the word problem is decidable). What can be said for the equations $x^n \leq x$ and $x^m \leq x \vee 1$, for $n \geq 3$ and $m \geq 2$?

7. In [5] and Section 6.1.2, it is shown that extensions of RL by expansive equations have an undecidable equational theory. Is this also true for other, non-expansive equations, such as $x^3 \leq x^2$ or $x \leq x^2 \vee 1$?

[2] We note that by Section 3.3, the varieties have the FMP.

8. All single-variable spinal equations are either knotted or of the form $x^n \leq x^m \vee 1$, where $n \neq m$. In the context of CRL, knotted equations have the FEP and thus have decidable universal theories. In Theorem 3.3.2, we establish that $x^n \leq x \vee 1$, where $n > 1$, has the FMP since its corresponding simple equation is completely linear. However, for equations $x^n \leq x^m \vee 1$, where $m > 1$, nothing is known. In particular, a running example in this thesis has been the simple equation

$$[\mathsf{d}] : x \leq x^2 \vee 1,$$

for which we have primarily stated negative results. By Proposition 3.1.3, $\mathsf{CRL} + [\mathsf{d}]$ does not have the FEP. By [23] and Section 4.4, the quasi-equational theory for $\mathsf{CRL} + [\mathsf{d}]$ does not have a primitive recursive decision procedure. On the other hand, by [14], such equations are known to have an undecidable the word problem for when extending RL. Is the quasi-equational theory of $\mathsf{CRL} + [\mathsf{d}]$ decidable or undecidable? Is the equational theory of $\mathsf{CRL} + [\mathsf{d}]$ decidable or undecidable?

BIBLIOGRAPHY

[1] G. Birkhoff. On the structure of abstract algebras. *Proc. Camb. Philos. Soc.*, 31(4):433–454, 1935.

[2] W. Blok and D. Pigozzi. Algebraizable logics. *Memoirs of the American Mathematical Society*, 77(396):1–78, 1989.

[3] W.J. Blok and C.J. van Alten. The finite embeddability property for residuated lattices, pocrims and bck-algebras. *Algebra Universalis*, 48:253–271, 2002.

[4] S. Burris and H.P. Sankappanavar. *A Course in Universal Algebra*, volume 91. Springer-Verlag, 01 1981.

[5] K. Chvalovský and R. Horčík. Full lambek calculus with contraction is undecidable. *Journal of Symbolic Logic*, 81(2):524–540, 2016.

[6] A. Ciabattoni, N. Galatos, and K. Terui. Algebraic proof theory for substructural logics: Cut-elimination and completions. *Annals of Pure and Applied Logic*, 163(3):266–290, 2012.

[7] A. Ciabottoni, B. Lellmann, C. Olarte, and E. Pimentel. From cut-free calculi to automated deduction: The case of bounded contraction. *Electronic Notes in Theoretical Computer Science*, 332:75 – 93, 2017. LSFA 2016 - 11th Workshop on Logical and Semantic Frameworks with Applications (LSFA).

[8] N. Galatos and P. Jipsen. Residuated frames with applications to decidability. *Transactions of the American Mathematical Societ*, 365(3):1219–1249, 2013.

[9] N. Galatos, P. Jipsen, T. Kowalski, and H. Ono. *Residuated Lattices: an algebraic glimpse at substructural logics*, volume 151. Elsevier, Amsterdam, 2007.

[10] N. Galatos and H. Ono. Algebraization, parameterized local deduction theorem and interpolation for substructural logics of **FL**. *Studia Logica*, 83:279303, 2006.

[11] G. Gentzen. Untersuchungen über das logische schließen I, II. *Mathematische Zeitschrift*, 39(1):95–135, 405–431, 1935.

[12] J. Hart, L. Rafter, and C. Tsinakis. The structure of commutative residuated lattices. *International Journal of Algebra and Computation*, 12:509–524, 2002.

[13] J. E. Hopcroft and J. D. Ullman. *Introduction To Automata Theory, Languages, And Computation*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1990.

[14] R. Horčík. Word problem for knotted residuated lattices. *Journal of Pure Applied Algebra*, 219:1548–1563, 2015.

[15] R. Horčík and K. Terui. Disjunction property and complexity of substructural logics. *Theoretical Computer Science*, 412(31):3992 – 4006, 2011.

[16] J. Lambek. How to program an infinite abacus. *Canadian Mathematical Bulletin*, 4:265–302, 1961.

[17] P. Lincoln, J. Mitchell, A. Scedrov, and N. Shankar. Decision problems for propositional linear logic. *Annals of Pure and Applied Logic*, 56:239–311, 1992.

[18] M. Lothaire. *Algebraic Combinatorics on Words*. Cambridge University Press, Cambridge, 2002.

[19] E. W. Mayr and A.R. Meyer. The complexity of the finite containment problem for petri nets. *Journal of the Association for Computing Machinery*, 28(3):561–576, 1981.

[20] M. Minksy. Recursive unsolvability of post's problem of 'tag' and other topics in the theory of turing machines. *The Annals of Mathematics*, 74:437–455, 1961.

[21] S. Roman. *Positive Solutions to Linear Systems: Convexity and Separation*. Springer New York, New York, NY, 2005.

[22] A. Tarski. A remark on functionally free algebras. *Annals of Mathematics*, 47(1):163–166, 1946.

[23] A. Urquhart. The complexity of decision procedures in relevance logic II. *Journal of Symbolic Logic*, 64(4):1774–1802, 1999.

[24] C.J. van Alten. The finite model property for knotted extensions of propositional linear logic. *Journal of Symbolic Logic*, 70(1):84–98, 2005.