The Journal of Symbolic Logic
Volume 87, Number 3, September 2022

MOST SIMPLE EXTENSIONS OF $\mathbf{FL_e}$ ARE UNDECIDABLE

NIKOLAOS GALATOS AND GAVIN ST. JOHN

Abstract. All known structural extensions of the substructural logic $\mathbf{FL_e}$, the Full Lambek calculus with exchange/commutativity (corresponding to subvarieties of commutative residuated lattices axiomatized by $\{\vee, \cdot, 1\}$-equations), have decidable theoremhood; in particular all the ones defined by knotted axioms enjoy strong decidability properties (such as the finite embeddability property). We provide infinitely many such extensions that have undecidable theoremhood, by encoding machines with undecidable halting problem. An even bigger class of extensions is shown to have undecidable deducibility problem (the corresponding varieties of residuated lattices have undecidable word problem); actually with very few exceptions, such as the knotted axioms and the other prespinal axioms, we prove that undecidability is ubiquitous. Known undecidability results for non-commutative extensions use an encoding that fails in the presence of commutativity, so and-branching counter machines are employed. Even these machines provide encodings that fail to capture proper extensions of commutativity, therefore we introduce a new variant that works on an exponential scale. The correctness of the encoding is established by employing the theory of residuated frames.

§1. Introduction. Substructural logics are defined as extensions of the Full Lambek calculus **FL** and include among others classical, intuitionistic, linear, relevance, bunched-implication, and many-valued logics. They find applications to areas as diverse as mathematical linguistics, philosophy, management of pointers in computer architecture, engineering, theoretical physics, and functional programming. Their algebraic semantics, in the sense of Blok and Pigozzi [1], are (pointed) residuated lattices (or FL-algebras) and they have an independent history with connections to classical and to ordered algebra. In particular they include the lattice of ideals of rings, lattice-ordered groups, algebras of relations, and of course Boolean and Heyting algebras. Pointed residuated lattices form a variety FL and its subvarieties correspond to extensions of **FL** via a dual lattice isomorphism; algebraic and logical properties are tightly linked. See [6] for an introduction to the area.

Decidability questions are at the core of the study of logical systems and here we explore logics/varieties with structure rich enough to allow for encoding the computation of machines with undecidable halting problem. This yields undecidability results for the word problem, and hence also for the quasiequational theory, of these varieties (namely the deducibility relation for the logics) and sometimes even the undecidability of the equational theory of the varieties (i.e., the theoremhood in the corresponding logics).

Received May 1, 2020.
2020 *Mathematics Subject Classification*. Primary 03B47, Secondary 08B15, 03G10, 06F05.
Key words and phrases. substructural logic, full Lambek calculus, simple extensions, residuated lattice, equational theory, word problem, decidability.

0022-4812/22/8703-0013
DOI:10.1017/jsl.2021.46

The equational theory of FL is decidable and the same holds for many of its standard extensions such as $\mathsf{FL_e}$ (with exchange/commutativity: $xy = yx$), $\mathsf{FL_w}$ (with weakening/integrality: $x \leq 1$), $\mathsf{FL_{ei}}$ (with exchange and weakening), $\mathsf{FL_{ec}}$ (with exchange and contraction: $x \leq x^2$), and $\mathsf{FL_{em}}$ (with exchange and mingle: $x^2 \leq x$). The equational theory of $\mathsf{FL_c}$ is one of the few known to be undecidable [3]; a precursor to this result is the fact that the equational theory of $\mathsf{FL_{ec}}$, even though decidable, is not primitive recursive [16]. The only other known subvarieties of FL with undecidable equational theory are the ones axiomatized by $x^m \leq x^n$ (where $0 < m < n$), the one axiomatized by the modular law, and the one axiomatized by commutativity, involutivity, and distributivity (corresponding to the relevance logic R). In particular, the last one is the only subvariety of $\mathsf{FL_e}$ with undecidable equational theory; actually distributivity does not correspond to a sequent structural rule, unless the syntax is expanded, so it is not even a structural extension of **FL**. On the contrary, prominent subvarieties of $\mathsf{FL_e}$, such as (proper, non-trivial) subvarieties axiomatized by any knotted inequality $x^m \leq x^n$ (where $m \neq n$), not only have a decidable equational theory but also a decidable quasiequational theory, and even the finite embeddability property [17]. (In [2] it is further shown that this remains true even under conditions weaker than commutativity.)

In contrast to the above, in this paper we construct infinitely many subvarieties of $\mathsf{FL_e}$ with undecidable equational theory. We also show that an even bigger collection of subvarieties of $\mathsf{FL_e}$ have an undecidable quasiequational theory, actually undecidable word problem. The encoding used for the undecidability of the word problem for $\mathsf{FL_e}$ does not work for its subvarieties, so we modify it in a novel way, by storing the values in the counter machines as powers of a sufficiently large constant, which depends on the subvariety.

A *residuated lattice* $\mathbf{R} = (R, \vee, \wedge, \cdot, \backslash, /, 1)$ is an algebraic structure such that $(R, \vee, \wedge)$ is a lattice, $(R, \cdot, 1)$ is a monoid, and the *law of residuation* holds: for all $x, y, z \in R$,

$$x \cdot y \leq z \text{ iff } x \leq z/y \text{ iff } y \leq x\backslash z,$$

where $\leq$ is the induced lattice order. The residuated lattice $\mathbf{R}$ is called *commutative* if $(R, \cdot, 1)$ is a commutative monoid; in such a case $x\backslash y = y/x$ for all $x, y \in R$ and we will use the notation $x \to y := x\backslash y$. It is well known that (commutative) residuated lattices form a variety denoted $\mathsf{(C)RL}$, see [6]. FL-algebras are defined as expansions of residuated lattices by an arbitrary constant 0 (which is used to define the negation operation), but we will not be making use of this constant in our encodings and our results remain true in the presence or absence of this constant.

For example, the extension of (0-free) $\mathbf{FL_e}$ with the structural rule

$$\frac{\Gamma, \Delta, \Delta, \Sigma \vdash \Pi \quad \Gamma, \Delta, \Delta, \Delta, \Sigma \vdash \Pi}{\Gamma, \Delta, \Sigma \vdash \Pi}\ (\varepsilon)$$

corresponds to the subvariety of CRL defined by the inequality $\varepsilon : x \leq x^2 \vee x^3$. More generally, structural rules, such as the above, correspond to inequalities in the signature $\{\vee, \cdot, 1\}$. We will show that theoremhood for $\mathbf{FL_e} + (\varepsilon)$ and the equational theory of $\mathsf{CRL} + \varepsilon$ are undecidable.

To establish the main undecidability results, we encode in the theory of commutative residuated lattices the computation of machines with undecidable halting problem; these are *and-branching counter machines*, a variant of counter machines, introduced in [11]. In Section 3 we outline their structure and in Section 4 we use them to give a simple and direct proof of the undecidability of every variety between CRL and RL, extending the result that was known for just these two varieties. This is done to set the stage for the much more complicated development that follows, while still introducing two of the main tools: abstract machines and residuated frames. The theory of residuated frames, developed in [5], is used to prove the completeness of the encoding, inspired by [3, 8]. The encodings used in the latter and the standard encodings, however, do not work in the presence of commutativity and this is why we make use of the one considered in [11]; in the Conclusions section (Section 9) we compare some of the encodings and discuss further some related results.

In the beginning of Section 5 we review the algebraic counterparts of structural rules: equalities in the signature $\{\vee, \cdot, 1\}$, as alluded to above, and show how they can be viewed as conjunctions of simple inequalities. We also explain why the encoding of and-branching counter machines fails to work for subvarieties of CRL axiomatized by such equations, thus necessitating the use of a novel encoding. Parts of Sections 5 and 6 explore properties of the new encoding that are needed for capturing the additional equations: the application of the equation even though it may interfere with the computation of the new machine, should not add any more accepted configurations. We refer to this condition as *admissibility* of the equation relative to the machine and also we work out some motivating examples.

In Section 7 we define the new machines as variants of and-branching counter machines working on an exponential level; the main idea is that the new equation does affect the computation but only in a linear/polynomial way, so if the encoding is done at the exponential level no new accepted configurations will be added. The desired properties of these exponential machines are studied in detail in Section 7 and a technical condition $(\star\star)$ emerges as a prominent condition for the given equation so that the encoding will work. In the beginning of Section 8 it becomes clear that if the added equation fails condition $(\star\star)$, not even the new exponential encoding will work to establish undecidability and thus such equations are outside the scope of this paper. However, in Section 7 it is shown that if the equation satisfies condition $(\star\star)$, then it indeed defines a variety of residuated lattices with undecidable word problem. Actually, Theorem 7.17 shows that almost all 1-variable equations actually satisfy condition $(\star\star)$, indicating that this condition is not really restrictive. In parallel to all this, starting already from Section 5 we introduce a very big class of equations, called *spineless*; the class is so big that we define it via its complement (prespinal equations), which forms a very small portion of all equations and admits a very natural and simple definition.

In Section 8 we show that, surprisingly, the very transparently defined spineless equations are precisely the ones that satisfy the technical condition $(\star\star)$. Therefore, the results of Sections 7 and 8 together imply that every spineless equation defines a variety with undecidable word problem (Corollary 8.14). Wanting to showcase this result as early as possible, we (re)state it in advance in Section 5 and use it to derive the stronger result of the undecidability of the equational theory for certain subvarieties; this is done by using a special form of the deduction theorem, relying on the characterization of congruences in commutative and expansive residuated lattices. In that sense, someone who reads just the first third of the paper, up to Section 5, has a full and clear grasp of all the notions and also knows the statements of the two main results of the paper, Theorem 5.5 and Theorem 5.9.

The proof, given in Section 8, that these two differently looking notions coincide is quite involved and relies on positive linear algebra, therefore Section 8 uses very different tools than the ones of the rest of the paper. We even find it helpful to move from the multiplicative notation for monoids, used in most of the paper and which suits the link to the encoding of the machines, to additive notation for monoids, giving rise to positive linear transformations and matrices with natural numbers as entries. Therefore we chose to have this proof done as the last section of the paper, as it already takes up one-fourth of the length of the paper.

§2. Preliminaries. We denote the sets of natural numbers, positive integers, and real numbers by $\mathbb{N}$, $\mathbb{Z}^+$, and $\mathbb{R}$, respectively. We denote the *powerset* of a set X by $\wp(X)$. Given a set X and a binary operation symbol $\cdot$, we denote by $(X^*, \cdot, 1)$ the free commutative monoid generated by X with unit 1. A *substitution* on X is a monoid homomorphism $\boldsymbol{\sigma} : X^* \to X^*$; substitutions are determined by their restriction to X. For $x \in X^*$, we write x^n to denote 1, if $n = 0$, and the term $x \cdot \dots \cdot x$ consisting of n copies of x for $n > 0$. For subsets A, B of X, we define $A \cdot B = \{a \cdot b : a \in A, b \in B\}$, and if $a \in X$ then $a \cdot B = \{a\} \cdot B$.

Let $\mathcal{L}$ be an algebraic language, i.e., containing no relational symbols. Given a set of variables X, $T(X)$ denotes the set of terms over X and $\mathcal{L}$ and $\mathbf{T}(X)$ the absolutely free algebra of terms. A *quasiequation* is (the universal closure of) a formula of the form

$$s_1 = t_1 \,\&\cdots\&\, s_n = t_n \implies s_0 = t_0, \tag{1}$$

where $s_0, t_0, \dots, s_n, t_n \in T(X)$ are terms and $n \in \mathbb{N}$. If $n = 0$ then the left-hand side is empty and we obtain an equation.

For a class of algebras $\mathcal{K}$ in the language $\mathcal{L}$, we say that (1) holds in $\mathcal{K}$ (i.e., $\mathcal{K} \models$ (1)) if for every algebra $\mathbf{A} \in \mathcal{K}$ and homomorphism $h : \mathbf{T}(X) \to \mathbf{A}$,

$$(\forall i \in \{1, \dots, n\})(\mathbf{A}, h \models s_i = t_i) \implies \mathbf{A}, h \models s_0 = t_0.$$

Here $\mathbf{A}, h \models s = t$ means $h(s) =_{\mathbf{A}} h(t)$.

A *presentation* 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 of algebras $\mathcal{V}$ 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 on input $s, t \in T(X)$ decides whether the quasiequation

$$\&E \implies s = t \tag{2}$$

holds in $\mathcal{V}$. Note that if $\mathcal{V}$ has undecidable word problem then its quasiequational and universal theories are undecidable as well. The word problem is also referred to as the local word problem and the quasiequational theory as the global word

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

problem, making explicit that in the former the antecedent of the quasiequations is fixed while in the latter it is unrestricted.

In the following the proofs of the undecidability of the word problem will rely on the fact that residuated lattices have a $\{\vee, \cdot, 1\}$-reduct, and therefore the undecidability results apply to all reducts that contain it.

§3. Counter machines. For proving undecidability we use a type of abstract machine known as an *And-branching k-Counter Machine* (k-ACM), introduced in [11], as they have an undecidable halting problem. A k-ACM is a tuple $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P}, q_f)$ representing a type of parallel-computing counter machine, where

- $\mathtt{R}_k := \{\mathtt{r}_1, \dots, \mathtt{r}_k\}$ is a set of k *registers*, each able to store a nonnegative 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.

A *configuration* $\mathtt{C}$ of a k-ACM is a tuple consisting of a single state and, for each register, a nonnegative integer indicating the contents of that register. We can imagine a configuration being a box labeled by a state and containing tokens each labelled by an element of the set $\mathtt{R}_k$. In essence, a configuration is specified by the state label and the multiset of register labels of the tokens. Since the order of the symbols is irrelevant, we represent a configuration $\mathtt{C}$ as a term in the free commutative monoid generated by $\mathtt{Q} \cup \mathtt{R}_k$, and canonically arranged as

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

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 $i = 1, ..., k$; if $n_i = 0$, we say the register $\mathtt{r}_i$ is *empty*. Since $\mathtt{C}$ contains precisely one state, we may define the set of configurations by $\mathtt{Conf(M)} := \mathtt{Q} \cdot \mathtt{R}_k^*$.

The instructions of a k-ACM replace a single configuration by a new configuration (via increment and decrement), or by two configurations (via forking). An *increment* instruction can be understood as "if a box is labeled by state q, add one register-$\mathtt{r}_i$ token and relabel the box with state q'," *decrement* as "if a box is labeled by state q, and $\mathtt{r}_i$ is not empty, remove one register-$\mathtt{r}_i$ token and relabel the box with state q'," and *forking* 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 is an element

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

of the free commutative semigroup $\mathtt{ID(M)}$ generated by $\mathtt{Conf(M)}$; we denote the associated binary operation by $\vee$.

In this way, we view $\mathtt{ID(M)}$ as a subset of the commutative semiring $\mathbf{A}_\mathtt{M} = (A_\mathtt{M}, \vee, \cdot, \bot, 1)$ generated by $\mathtt{R}_k \cup \mathtt{Q}$, where

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

- $(A_{\mathtt{M}}, \vee, \bot)$ is a commutative monoid with additive identity $\bot$,
- $(A_{\mathtt{M}}, \cdot, 1)$ is a commutative monoid with multiplicative identity 1, and
- multiplication distributes over join.

Even though $\vee$ in $\mathbf{A}_{\mathtt{M}}$ is not defined to be idempotent, we will consider homomorphisms from $\mathbf{A}_{\mathtt{M}}$ that will map $\vee$ to a semilattice operation and for our applications it would not hurt to define $\vee$ to be idempotent. However, the non-idempotent status of ID's matches better the intuition of computation and this is the reason for our choice.

Since multiplication fully distributes over $\vee$ in $\mathbf{A}_{\mathtt{M}}$, each element of $A_{\mathtt{M}}$ can be written as a finite join $\bigvee_{i \in I} m_i$, where I is a finite (possibly empty) set, of monoid terms $m_i \in (\mathtt{Q} \cup \mathtt{R}_k)^*$, for all $i \in I$; recall that the join of the empty set is the bottom element ($\bot = \bigvee \emptyset$). As usual, each element of $\mathbf{A}_{\mathtt{M}}$, which is the equivalence class $[t]$ of a term t in the absolutely free algebra over $\{\vee, \cdot, 1\}$ and $\mathtt{Q} \cup \mathtt{R}_k$, will be identified with the term t itself, when no confusion arises.

Formally, an instruction p of a k-ACM is an expression of the form $q \leq q'\mathtt{r}_i$, $q\mathtt{r}_i \leq q'$, or $q \leq q' \vee q''$, where $q, q', q'' \in \mathtt{Q}$ and $\mathtt{r}_i \in \mathtt{R}_k$, representing *increment* $\mathtt{r}_i$, *decrement* $\mathtt{r}_i$, and *fork*, respectively. We will often write $p : \mathtt{C} \leq \mathtt{u}$ to indicate the instruction p is given by $\mathtt{C} \leq \mathtt{u}$, where $\mathtt{C} \in \mathtt{Conf(M)}$ and $\mathtt{u} \in \mathtt{ID(M)}$. For a state $q \in \mathtt{Q}$, we say p is a *q-instruction* if $p : qx \leq \mathtt{u}$ for some $x \in \mathtt{R}_k^*$. Note that a machine $\mathtt{M}$ with final state q_f contains no q_f-instructions by definition.

The *computation relation* $\leq$ for the machine $\mathtt{M} = (\mathtt{R}_K, \mathtt{Q}, \mathtt{P}, q_f)$ is defined to be the smallest $\{\cdot, \vee\}$-compatible preorder containing $\mathtt{P}$, and will be denoted by $\leq_{\mathtt{M}}$. For a given instruction $p : \mathtt{C} \leq \mathtt{u}$, it will be useful to define the relation $\leq^p$ to be the closure of p under the inference rules

$$\frac{v \leq^p w}{vx \leq^p wx}\ [\cdot] \quad \text{and} \quad \frac{v \leq^p w}{v \vee t \leq^p w \vee t}\ [\vee],$$

for all $v, w, x, t \in A_{\mathtt{M}}$ (in that order, without loss of generality, due to the distributivity of $\cdot$ over $\vee$). Consequently, $v \leq^p w$ if and only if $v = \mathtt{C}x \vee t$ and $w = \mathtt{u}x \vee t$, for some $x, t \in A_{\mathtt{M}}$; these equalities are understood inside $A_{\mathtt{M}}$, so the terms v and $\mathtt{C}x \vee t$ need not be identical. We therefore conclude that if $v \leq^p w$, then $v \in \mathtt{ID(M)}$ if and only if $w \in \mathtt{ID(M)}$.

It is easily verified that $\leq_{\mathtt{M}}$ is equivalent to the smallest preorder generated by $\bigcup\{\leq^p : p \in \mathtt{P}\}$. Therefore, if $s \leq_{\mathtt{M}} t$, then there exist $n \in \mathbb{N}$, a sequence of $A_{\mathtt{M}}$-terms $t_0, \dots, t_n$ and a sequence of instructions $p_1, \dots, p_n$ from $\mathtt{P}$, collectively called a *computation in* $\mathtt{M}$ of *length* n witnessing $s \leq_{\mathtt{M}} t$, such that

$$s_0 =_{\mathbf{A}_{\mathtt{M}}} t_0 \leq^{p_1} t_1 \leq^{p_2} \cdots \leq^{p_n} t_n =_{\mathbf{A}_{\mathtt{M}}} t.$$

Clearly, if there is a computation witnessing $s \leq_{\mathtt{M}} t$, then there is a computation of minimal length, the value of which we simply call the *computation length*. The following result is an easy consequence of the definitions.

Lemma 3.1. *Let $s, t, t' \in A_{\mathtt{M}}$.*

1. *If $s \leq_{\mathtt{M}} t$, then there exists a computation witnessing it and furthermore, $s \in \mathtt{ID(M)}$ iff $t \in \mathtt{ID(M)}$.*

2. *$t \vee t' \leq_{\mathtt{M}} s$ if and only if there exist $s', s'' \in A_{\mathtt{M}}$ such that $s = s' \vee s''$, $t \leq_{\mathtt{M}} s'$, and $t' \leq_{\mathtt{M}} s''$. Furthermore, the sum of the computation lengths of $t \leq_{\mathtt{M}} s'$ and $t' \leq_{\mathtt{M}} s''$ is less than or equal to the computation length of $t \vee t' \leq_{\mathtt{M}} s$.*

An ID consisting entirely of configurations labeled by the state q_f with all registers empty is called a *final ID* and we denote the set of final ID's by $\mathtt{Fin}(\mathtt{M}) := \{\bigvee_{i=1}^{n} q_f : n \geq 1\}$. Our choice to not assume the idempotency of $\vee$ in $\mathbf{A}_{\mathtt{M}}$ explains the necessity of treating $\bigvee_{i=1}^{n} q_f$ as a final ID. We say that a term $t \in A_{\mathtt{M}}$ is *accepted by* $\mathtt{M}$ if there exists $\mathtt{u}_f \in \mathtt{Fin}(\mathtt{M})$ such that $t \leq_{\mathtt{M}} \mathtt{u}_f$, and we define the set of *accepted terms* to be $\mathrm{Acc}(\mathtt{M}) := \{t \in A_{\mathtt{M}} : \exists \mathtt{u}_f \in \mathtt{Fin}(\mathtt{M}),\ t \leq_{\mathtt{M}} \mathtt{u}_f\}$.

THEOREM 3.2. [10, 11, 13] *There exists a 2-ACM $\tilde{\mathtt{M}}$ such that membership in Acc$(\tilde{\mathtt{M}})$ is undecidable.*

EXAMPLE 3.3. Consider the 1-ACM $\mathtt{M}_{\text{even}} = (\mathtt{R}_1, \mathtt{Q}_{\text{even}}, \mathtt{P}_{\text{even}}, q_f)$, where $\mathtt{Q}_{\text{even}} = \{q_0, q_1, q_f\}$ and $\mathtt{P}_{\text{even}} = \{p_0, p_1, p_f\}$ is given by

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

For example, $q_0\mathtt{r}_1^2 \leq^{p_0} q_1\mathtt{r}_1 \leq^{p_1} q_0 \leq^{p_f} q_f \vee q_f$ is a computation showing that $q_0\mathtt{r}_1^2$ is accepted. On the other hand $q_0\mathtt{r}_1 \leq^{p_0} q_1$ and $q_0\mathtt{r}_1 \leq^{p_f} (q_f \vee q_f)\mathtt{r}_1 = q_f\mathtt{r}_1 \vee q_f\mathtt{r}_1$ are the only maximal computations starting with $q_0\mathtt{r}_1$ and none of them ends in a final configuration, so $q_0\mathtt{r}_1$ is not accepted. In general, it is easy to see that $q_0\mathtt{r}_1^n \in \mathrm{Acc}(\mathtt{M}_{\text{even}})$ if and only if n is even.

LEMMA 3.4. *Let $\mathtt{M}$ be an ACM.*

1. $\mathrm{Acc}(\mathtt{M}) \subseteq \mathtt{ID}(\mathtt{M})$ *and the terms in any computation ending in a final ID are all ID's.*
2. *For all $u, v \in A_{\mathtt{M}}$, $u \vee v \in \mathrm{Acc}(\mathtt{M})$ if and only if $u \in \mathrm{Acc}(\mathtt{M})$ and $v \in \mathrm{Acc}(\mathtt{M})$.*

PROOF. The first claim follows from Lemma 3.1(1) by induction on the computation length since $\mathtt{Fin}(\mathtt{M}) \subseteq \mathtt{ID}(\mathtt{M})$ by definition. The second claim follows from Lemma 3.1(2) since $\mathtt{Fin}(\mathtt{M})$ is exactly the set of finite non-empty joins of the same configuration q_f. ⊣

Lemma 3.4 shows that in a computation witnessing the acceptance of an ID all configurations are ID's and therefore for those cases the inference rule [$\cdot$] could have been restricted to $x \in \mathtt{R}_k^*$.

§4. Machines and residuated frames. As a demonstration of our general technique, we will use counter machines and residuated frames to show that any variety between CRL and RL has an undecidable word problem.

Let $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P}, q_f)$ be an ACM. For each $u \in \mathtt{ID}(\mathtt{M})$, formally viewed as an $\{\vee, \cdot, 1\}$-term in $\mathbf{T}(\mathtt{R}_k \cup \mathtt{Q})$, we define the quasiequation $\mathrm{acc}_{\mathtt{M}}(u)$ to be

$$\& \mathtt{P}_{\text{com}} \Rightarrow u \leq q_f,$$

where q_f is the final state of $\mathtt{M}$, and $\mathtt{P}_{\text{com}} := \mathtt{P} \cup \{xy \leq yx : x, y \in \mathtt{R}_k \cup \mathtt{Q}\}$ is the (finite) set of instructions $\mathtt{P}$ together with a finite set encoding commutativity for

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

letters (and hence also all words) over the set $\mathtt{R}_k \cup \mathtt{Q}$; for our purposes we could actually restrict the x in $\mathtt{P}_{\mathrm{com}}$ to only state variables.

The following lemma shows that computations in machines can be performed also in the theory of residuated lattices.

LEMMA 4.1. *Let* $\mathtt{M}$ *be an ACM and* u *an ID. If* $u \in \mathrm{Acc}(\mathtt{M})$ *then* $\mathsf{RL} \models \mathrm{acc}_{\mathtt{M}}(u)$.

PROOF. Let $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P}, q_f)$ be an ACM and suppose $u \in \mathtt{ID}(\mathtt{M})$ is accepted in $\mathtt{M}$. We proceed by induction on the length n of the computation witnessing the acceptance of u in $\mathtt{M}$. If the length is zero then $u \in \mathtt{Fin}(\mathtt{M})$. Since $\vee$ is idempotent in residuated lattices, $\mathsf{RL} \models \mathtt{u}_f \leq q_f$ for any $\mathtt{u}_f \in \mathtt{Fin}(\mathtt{M})$. Hence $\mathsf{RL} \models \mathrm{acc}_{\mathtt{M}}(u)$ *a fortiori*. Now, suppose the claim holds for all accepted ID's with computation length $0 \leq k < n$. By Lemma 3.4(1), there is an instruction $p \in \mathtt{P}$ such that $u \leq^p u' \in \mathrm{Acc}(\mathtt{M})$ for some $u' \in \mathtt{ID}(\mathtt{M})$, where the acceptance computation of u' has length less than n. Formally viewing u' as an element in $\mathbf{T}(X)$ where $X = \mathtt{R}_k \cup \mathtt{Q}$, $\mathsf{RL} \models \mathrm{acc}_{\mathtt{M}}(u')$ by the induction hypothesis.

Now, suppose that for a residuated lattice $\mathbf{R}$ and for a homomorphism $f : \mathbf{T}(X) \to \mathbf{R}$ we have $\mathbf{R}, f \models \mathtt{P}_{\mathrm{com}}$. Hence $f(u') \leq_{\mathbf{R}} f(q_f)$ since $\mathsf{RL} \models \mathrm{acc}_{\mathtt{M}}(u')$. As $\leq_{\mathbf{R}}$ is transitive, we need only show $f(u) \leq_{\mathbf{R}} f(u')$ to establish $\mathbf{R}, f \models \mathrm{acc}_{\mathtt{M}}(u)$.

Let $\mathbf{S}(X)$ be the free algebra over $\{\vee, \cdot, 1\}$ and X. As $\mathbf{R}$ has a semiring reduct and $f(a)f(b) =_{\mathbf{R}} f(b)f(a)$, for all $a, b \in X$, the restriction of f on $\mathbf{S}(X)$ factors through $\mathbf{A}_{\mathtt{M}}^{+} := \mathbf{A}_{\mathtt{M}} \setminus \{\bot\}$ as $f : S(X) \xrightarrow{v} A_{\mathtt{M}}^{+} \xrightarrow{h} R$ as a semiring homomorphism, where v is the natural surjective homomorphism and h is a semiring homomorphism. So, $h(a)h(b) =_{\mathbf{R}} h(b)h(a)$, for all $a, b \in X$, and $h(\mathtt{C}) \leq_{\mathbf{R}} h(\mathtt{v})$ where $p : \mathtt{C} \leq \mathtt{v}$. By definition of $\leq^p$,

$$u =_{\mathbf{A}_{\mathtt{M}}} \mathtt{C}x \vee w \leq^p \mathtt{v}x \vee w =_{\mathbf{A}_{\mathtt{M}}} u',$$

for some $x \in X^*$ and $w \in A_{\mathtt{M}}$, where $\mathtt{v}x \vee w =_{\mathbf{A}_{\mathtt{M}}} \mathtt{v}x$ if $w = \bot$. Using the properties above and the fact that h is a semiring homomorphism we obtain

$$h(u) =_{\mathbf{R}} h(\mathtt{C}x \vee w) \leq_{\mathbf{R}} h(\mathtt{v}x \vee w) =_{\mathbf{R}} h(u').$$

It follows that $h(u) \leq_{\mathbf{R}} h(u')$ and therefore that $f(u) \leq_{\mathbf{R}} f(u')$. ⊣

To show the converse of Lemma 4.1, we will need to show that given an ACM $\mathtt{M}$, if $u \notin \mathrm{Acc}(\mathtt{M})$ then there is a residuated lattice $\mathbf{W}_{\mathtt{M}}^{+}$ (which will actually even be commutative) that falsifies $\mathrm{acc}_{\mathtt{M}}(u)$; actually, in the proof we proceed by contraposition. We will further prove that every subvariety of RL that contains $\mathbf{W}_{\tilde{\mathtt{M}}}^{+}$ has undecidable word problem (and thus undecidable quasiequational theory). The construction of $\mathbf{W}_{\mathtt{M}}^{+}$ is based on residuated frames [5], structures that will also be used later in the paper, so we define them briefly here.

For the purposes of this paper, a *commutative residuated frame* is a structure $\mathbf{W} = (W, W', N, \cdot, 1)$, where $(W, \cdot, 1)$ is a commutative monoid, W' is a set, and N is a subset of $W \times W'$, such that there exists a function $/\!\!/ : W' \times W \to W'$ with: $\forall x, y \in W\ z \in W'$, $x \cdot y \, N \, z$ iff $x \, N \, z /\!\!/ y$. Given such a residuated frame, for $X \subseteq W$, $x \in W$, $Y \subseteq W'$, and $y \in W'$, we define $X \, N \, y$ to mean $x \, N \, y$ for all $x \in X$, and $x \, N \, Y$ to mean $x \, N \, y$ for all $y \in Y$. For $X \subseteq W$ and $Y \subseteq W'$, we define $X^{\rhd} := \{y \in W' : X \, N \, y\}$, $Y^{\lhd} := \{x \in W : x \, N \, Y\}$. The pair $(^{\rhd}, ^{\lhd})$ forms what is

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

known as a *Galois connection*, and we will make use of the fact that $X_1^{\rhd\lhd} \subseteq X_2^{\rhd\lhd}$ if and only if $X_2^{\rhd} \subseteq X_1^{\rhd}$ for any $X_1, X_2 \subseteq W$.

Define $\gamma(X) = X^{\rhd\lhd}$. We write $\gamma(x) = \gamma(\{x\})$ for $x \in W$, and $\wp(W)_\gamma = \gamma[\wp(W)]$. It follows from [5] that the algebra $\mathbf{W}^+ := (\wp(W)_\gamma, \cap, \cup_\gamma, \cdot_\gamma, \to, \gamma(1))$ is a commutative residuated lattice, where $X \cup_\gamma Y := \gamma(X \cup Y)$, $X \cdot_\gamma Y := \gamma(X \cdot Y)$, and $X \to Y := \{z \in W : X \cdot \{z\} \subseteq Y\}$.

Inspired by [8], given an ACM $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P}, q_f)$ we define the tuple $\mathbf{W}_\mathtt{M} = (W_\mathtt{M}, W_\mathtt{M}, N_\mathtt{M}, \cdot, 1)$, where $W_\mathtt{M} := (\mathtt{Q} \cup \mathtt{R}_k)^* \subseteq A_\mathtt{M}$ and $x\, N_\mathtt{M}\, y$ if and only if $xy \in \mathrm{Acc}(\mathtt{M})$, for all $x, y \in W_\mathtt{M}$.

Lemma 4.2. *$\mathbf{W}_\mathtt{M}$ is a residuated frame and therefore $\mathbf{W}_\mathtt{M}^+ \in$ CRL.*

Proof. We define $z \,/\!/\, y = yz$. Clearly, for $x, y, z \in W_\mathtt{M}$, $xy\, N_\mathtt{M}\, z$ iff $xyz \in \mathrm{Acc}(\mathtt{M})$ iff $x\, N_\mathtt{M}\, yz$. ⊣

For an ACM $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P}, q_f)$, we define the assignment $e : \mathtt{Q} \cup \mathtt{R}_k \to W_\mathtt{M}^+$ via $e(a) := \{a\}^{\rhd\lhd}$ and its homomorphic extension $\bar{e} : \mathbf{T}(\mathtt{Q} \cup \mathtt{R}_k) \to \mathbf{W}_\mathtt{M}^+$.

We will need to make use of the following technical lemma.

Lemma 4.3. *If $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P}, q_f)$ is an ACM, then $\mathbf{W}_\mathtt{M}^+, \bar{e} \models \mathtt{P}_{\mathrm{com}}$. Furthermore, $\bar{e}(x \vee y) = \{x, y\}^{\rhd\lhd}$ for any $x, y \in W_\mathtt{M}$.*

Proof. In [5] it is shown that the map γ satisfies the properties $\gamma(\gamma(X) \cdot \gamma(Y)) = \gamma(X \cdot Y)$ and $\gamma(\gamma(X) \cup \gamma(Y)) = \gamma(X \cup Y)$, for all $X, Y \subseteq W$. Using the first one, for each $a, b \in \mathtt{Q} \cup \mathtt{R}_k$ we have

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

It follows by induction that $\bar{e}(x) = \gamma(x)$ for each $x \in W_\mathtt{M}$.

Now, let $x, y \in W_\mathtt{M}$. 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 follows from the second property of γ above.

Since $\mathbf{W}_\mathtt{M}^+$ is commutative, we need only show $\mathbf{W}_\mathtt{M}^+ \models \mathtt{P}$. Let $p : \mathtt{C} \leq \mathtt{C}_1 \vee \mathtt{C}_2$ be in $\mathtt{P}$.[1] By the calculation above, $\bar{e}(\mathtt{C}) = \{\mathtt{C}\}^{\rhd\lhd}$ and $\bar{e}(\mathtt{C}_1 \vee \mathtt{C}_2) = \{\mathtt{C}_1, \mathtt{C}_2\}^{\rhd\lhd}$. So, to show $\mathbf{W}_\mathtt{M}^+, \bar{e} \models \mathtt{C} \leq \mathtt{C}_1 \vee \mathtt{C}_2$, we need to show that $\{\mathtt{C}\}^{\rhd\lhd} \subseteq \{\mathtt{C}_1, \mathtt{C}_2\}^{\rhd\lhd}$, or equivalently that $\{\mathtt{C}_1, \mathtt{C}_2\}^{\rhd} \subseteq \{\mathtt{C}\}^{\rhd}$. Suppose $x \in \{\mathtt{C}_1, \mathtt{C}_2\}^{\rhd}$, then $\mathtt{C}_1\, N_\mathtt{M}\, x$ and $\mathtt{C}_2\, N_\mathtt{M}\, x$, so $\mathtt{C}_1 x, \mathtt{C}_2 x \in \mathrm{Acc}(\mathtt{M})$. Now $\mathtt{C} \leq^p \mathtt{C}_1 \vee \mathtt{C}_2$ implies $\mathtt{C}x \leq^p (\mathtt{C}_1 \vee \mathtt{C}_2)x$, thus by Lemma 3.4(2)

$$\mathtt{C}x \leq^p (\mathtt{C}_1 \vee \mathtt{C}_2)x = \mathtt{C}_1 x \vee \mathtt{C}_2 x \in \mathrm{Acc}(\mathtt{M}),$$

and it follows that $\mathtt{C}\, N_\mathtt{M}\, x$, or equivalently $x \in \{\mathtt{C}\}^{\rhd}$. ⊣

Lemma 4.4. *Let $\mathcal{V}$ be a subvariety of* RL *containing $\mathbf{W}_\mathtt{M}^+$ for some ACM $\mathtt{M}$. Then for all $u \in \mathtt{ID}(\mathtt{M})$, $u \in \mathrm{Acc}(\mathtt{M})$ if and only if $\mathcal{V} \models \mathrm{acc}_\mathtt{M}(u)$.*

Proof. Let $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P}, q_f)$ be a k-ACM. The forward direction follows from Lemma 4.1. For the reverse direction note that from $\mathbf{W}_\mathtt{M}^+ \in \mathcal{V}$ we have $\mathbf{W}_\mathtt{M}^+ \models \mathrm{acc}_\mathtt{M}(u)$. By Lemma 4.3, $\mathbf{W}_\mathtt{M}^+, \bar{e} \models \mathtt{P}_{\mathrm{com}}$ and so $\mathbf{W}_\mathtt{M}^+, \bar{e} \models u \leq q_f$. Let $t_1, \ldots, t_n \in (\mathtt{Q} \cup \mathtt{R}_k)^*$ be given so that $u = t_1 \vee \cdots \vee t_n$, so $\bar{e}(t_1 \vee \cdots \vee t_n) \subseteq \bar{e}(q_f)$, which yields

[1]The argument that follows clearly works for $\mathtt{v} = \mathtt{C}_1$ as well.

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

$\{t_1, \dots, t_n\}^{\triangleright\triangleleft} \subseteq q_f{}^{\triangleright\triangleleft}$ by Lemma 4.3. This is equivalent to $\{q_f\}^{\triangleright} \subseteq \{t_1, \dots, t_n\}^{\triangleright}$. Since $\leq_{\mathtt{M}}$ is reflexive, $q_f \in \mathrm{Acc}(\mathtt{M})$ and thus $q_f \; N_{\mathtt{M}} \; 1$, so $1 \in \{q_f\}^{\triangleright}$. Therefore, $1 \in \{t_1, \dots, t_n\}^{\triangleright}$, that is $\{t_1, \dots, t_n\} \; N_{\mathtt{M}} \; 1$, so $t_1 \; N_{\mathtt{M}} \; 1, \dots, t_n \; N_{\mathtt{M}} \; 1$. Hence $t_1, \dots, t_n \in \mathrm{Acc}(\mathtt{M})$ and by Lemma 3.4(2) we conclude $u \in \mathrm{Acc}(\mathtt{M})$. $\dashv$

As a consequence of Lemma 4.4, if $\mathcal{V} \subseteq \mathsf{RL}$ is a variety containing $\mathbf{W}_{\mathtt{M}}^{+}$, for some ACM $\mathtt{M}$, then $\{\mathrm{acc}_{\mathtt{M}}(u) : u \in \mathrm{Acc}(\mathtt{M})\} = \{\mathrm{acc}_{\mathtt{M}}(u) : \mathcal{V} \models \mathrm{acc}_{\mathtt{M}}(u)\}$. Since $\langle \mathtt{Q} \cup \mathtt{R}_k, \mathrm{P}_{\mathrm{com}} \rangle$ is a finite presentation and all equations in $\mathrm{acc}_{\mathtt{M}}(u)$ have a common antecedent $\&\mathrm{P}_{\mathrm{com}}$, the following is immediate:

THEOREM 4.5. *Let* $\mathtt{M}$ *be an ACM and* $\mathbf{W}_{\mathtt{M}}^{+} \in \mathcal{V} \subseteq \mathsf{RL}$ *for a variety* $\mathcal{V}$. *Then deciding the word problem of* $\mathcal{V}$ *is at least as hard as deciding membership in* $\mathrm{Acc}(\mathtt{M})$.

COROLLARY 4.6. *If* $\mathcal{V}$ *is a subvariety of* RL *containing* $\mathbf{W}_{\mathtt{M}}^{+}$, *where* $\mathtt{M}$ *is an ACM such that membership in* $\mathrm{Acc}(\mathtt{M})$ *is undecidable, then* $\mathcal{V}$ *has an undecidable word problem. In particular, any variety in the interval* CRL *to* RL *has undecidable word problem since* $\mathbf{W}_{\tilde{\mathtt{M}}}^{+} \in \mathsf{CRL}$, *where* $\tilde{\mathtt{M}}$ *is the machine from Theorem 3.2.*

The above results hold even for the $\{\vee, \cdot, 1\}$ reducts of these varieties. Since $\{\mathrm{acc}_{\mathtt{M}}(u) : \mathcal{V} \models \mathrm{acc}_{\mathtt{M}}(u)\} \subseteq \{\xi : \xi \text{ is a quasieq. such that } \mathcal{V} \models \xi\}$, we therefore also obtain the undecidability of the quasiequational theory. The quasiequational theories of RL and CRL alone were known to be undecidable; see [9] and [11], respectively.

§5. Equations in the signature $\{\vee, \cdot, 1\}$ and machine admissibility. Our goal is to find proper subvarieties of $(\mathsf{C})\mathsf{RL}$ for which Theorem 4.5 will be applicable, as well as strengthening this result to the undecidability of the equational theory for some proper subvarieties of CRL. Since structural rules correspond to equations in the signature $\{\vee, \cdot, 1\}$ (see [5]), we will restrict our attention to varieties axiomatized by such equations.

Since in residuated lattices multiplication distributes over joins, every equation over $\{\vee, \cdot, 1\}$ is equivalent to an equality between finite joins of monoid terms. This equality can in turn be written as two inequalities and in each one of them the joins on the left-hand side of the inequality yield a conjunction of inequalities of the form $t_0 \leq t_1 \vee \cdots \vee t_l$, where $t_0, \dots, t_l \in X^*$ are monoid terms. We call equations of this form *basic equations* (or basic inequalities), if it is further true that the variable sets on the two sides of the inequality are the same. It can be easily shown that joinands on the right-hand side containing variables that do not appear on the left can be safely omitted, resulting in an equivalent equation. (In the case where all the joinands are of this form, the equation implies $1 \leq x$, so it defines the trivial variety) Furthermore, if there are variables that appear on the left and not on the right, then the inequality implies integrality ($x \leq 1$). These claims are easy to prove and the needed instantiations are also mentioned on page 277 of [4]. The trivial variety and variety axiomatized by integrality together with any set of $\{\vee, \cdot, 1\}$-equations have the FEP, hence decidable universal (and quasiequational) theory. Therefore the restriction of the variables appearing on both sides does not leave out any unknown cases of (un)decidability. Via a process of *linearization* [5] any basic equation is further equivalent to one where the term t_0 is *linear*, namely to an equation of the form $x_1 \cdots x_n \leq \bigvee_{i=1}^{k} m_i$, where $k \geq 1$, $m_1, \dots, m_k \in X^*$ and $x_1, \dots, x_n$ are distinct elements of X. (For example the equation $x^2 \leq x$ is equivalent to the linearized equation $x_1x_2 \leq x_1 \vee x_2$) Such equations are called simple in [5], but in this paper we will reserve the name *simple equations of* RL for the subclass where the variable sets on the left- and the right-hand sides of the equation are equal.

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 will informally use variables like x, y, z in some of the examples, as well. We also define $\mathbf{x}_n := (x_1, \ldots, x_n)$, for all $n \in \mathbb{Z}^+$ and for a tuple $a \in \mathbb{N}^n$ of natural numbers, we define $\mathbf{x}_n^a = x_1^{a(1)} \cdots x_n^{a(n)}$; we also define $\mathbf{x}_n^{\mathbf{1}} = x_1 \cdots x_n$. For reasons that will be clear later, we will actually think of a as a column vector (as opposed to a row vector).

For a simple equation ε, we will be interested in its commutative version $\varepsilon_{\mathcal{C}}$, obtained from ε by rearranging the variables within each monoid term according to the natural ordering of their indices and removing any resulting duplicate joinands on the right-hand side. In particular, $\mathsf{CRL} + \varepsilon \models \varepsilon_{\mathcal{C}}$, and we call equations of the form $\varepsilon_{\mathcal{C}}$ *simple equations of* CRL. As the encodings for the undecidability are harder for commutative varieties than for arbitrary ones, by proving the results in the commutative case we obtain as corollaries results for general subvarieties of RL. Therefore, we restrict ourselves to simple equations of CRL and we will refer to them simply as *simple equations*. Such equations are of the form

$$[\mathrm{D}] : \mathbf{x}_n^{\mathbf{1}} \leq \bigvee_{d \in \mathrm{D}} \mathbf{x}_n^d,$$

where D is a finite nonempty set of n-column vectors with entries in $\mathbb{N}$, such that the variable sets on the two sides are the same, namely there is no row such that all column vectors in D are zero on that row. Note that because of the equality of the variable sets on the left and on the right and due to the idempotency of join in residuated lattices, every simple equation is fully determined by the set of joinands on its right-hand side. Our notation is chosen so that if D is a set of n-columns, then [D] denotes the simple equation displayed above (the exponents of the joinands on the right-hand side come from D).

For the bigger class of basic equations of CRL (which may not be linearized), if D is again a nonempty set of column vectors and f a column vector over the positive integers, we denote by $[f, \mathrm{D}]$, the basic equation

$$[f, \mathrm{D}] : \mathbf{x}_n^f \leq \bigvee_{d \in \mathrm{D}} \mathbf{x}_n^d.$$

The following theorem provides a link between simple equations that hold in $\mathbf{W}^+$ and conditions that hold in $\mathbf{W}$.

Let $(W, \cdot, 1)$ be a commutative monoid and [D] a n-variable simple equation given by $\mathrm{D} = \{d_j : 1 \leq j \leq m\}$. If $\mathbf{W} = (W, W', N, \cdot, 1)$ is a residuated frame then we write $\mathbf{W} \models (\mathrm{D})$ iff for all $\mathbf{u}_n \in W^n$ and $v \in W'$, the following implication is satisfied (the premises above the line are understood conjunctively and the vertical

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

line denotes the implication to the conclusion below):

$$\frac{\mathbf{u}_n^{d_1}\, N\, v \quad \cdots \quad \mathbf{u}_n^{d_m}\, N\, v}{\mathbf{u}_n^{\mathbf{1}}\, N\, v}\ (\mathrm{D})$$

For example if [D] is $x_1 x_1 \leq x_1 \vee x_2$, then (D) is:

$$\forall x_1, x_2 \in W, v \in W', x_1\, N\, v \,\&\, x_2\, N\, v \implies x_1 x_2\, N\, v.$$

THEOREM 5.1 [5]. *Let* [D] *be a simple equation and suppose* $\mathbf{W}$ *is a residuated frame. Then* $\mathbf{W}^+ \models$ [D] *iff* $\mathbf{W} \models (\mathrm{D})$.

5.1. Motivation for subvarieties of CRL**.** Recall from Example 3.3, that the computations of the 1-ACM $\mathtt{M}_{\mathrm{even}}$ leading to a final state are faithfully represented by the inequality relation of CRL, in the sense that $\mathsf{CRL} \models (\&\mathtt{P} \Rightarrow u \leq q_f)$ iff $u \in \mathrm{Acc}(\mathtt{M})$. If we consider the inequality relation in $\mathsf{CRL}_{\mathsf{s}}$, where s is the simple equation $x \leq x^2 \vee x^4$, we observe that for the computation relation of a machine to be faithfully represented by the associated inequality relation it must further admit the "ambient instruction" given by

$$t \leq^{\mathsf{s}} t^2 \vee t^4,$$

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 $\leq_{\mathsf{sM}_{\mathrm{even}}}$ be the smallest compatible preorder generated by $\mathtt{P}_{\mathrm{even}} \cup \leq^{\mathsf{s}}$, and define $\mathrm{Acc}(\mathsf{sM}_{\mathrm{even}})$ to be the set of accepted ID's under the relation $\leq_{\mathsf{sM}_{\mathrm{even}}}$. It is clear that $\mathrm{Acc}(\mathtt{M}_{\mathrm{even}}) \subseteq \mathrm{Acc}(\mathsf{sM}_{\mathrm{even}})$ since $\leq_{\mathtt{M}_{\mathrm{even}}} \subseteq \leq_{\mathsf{sM}_{\mathrm{even}}}$, and since there are no instructions (nor instances of $\leq^{\mathsf{s}}$) that remove state variables we obtain $\mathrm{Acc}(\mathsf{sM}_{\mathrm{even}}) \subseteq \mathtt{ID}(\mathtt{M}_{\mathrm{even}})$. However, while $q_0\mathtt{r}_1^3 \notin \mathrm{Acc}(\mathtt{M}_{\mathrm{even}})$, we have $q_0\mathtt{r}_1^3 \in \mathrm{Acc}(\mathsf{sM}_{\mathrm{even}})$ since

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

It is clear that the expansion of the machine by the ambient instruction (needed for representing the inequality relation in $\mathsf{CRL}_{\mathsf{s}}$) does not have the same computation relation, or put differently the machine $\mathtt{M}_{\mathrm{even}}$ is not suitable for representing the inequality relation in $\mathsf{CRL}_{\mathsf{s}}$ 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 $\tilde{\mathtt{M}}$) and in which these ambient instructions are available/admissible. For that reason we cannot use the same argumentation to show that $\mathsf{CRL}_{\mathsf{s}}$ has undecidable word problem.

Exactly the same issue occurs if the simple equation is contraction $\mathsf{c} : x \leq x^2$. 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 [17]. However, we will show that even though for the equation s above the current encoding is problematic (as is with contraction), surprisingly, unlike with contraction, there is a different encoding that works for s; this will allow us to prove that the word problem for $\mathsf{CRL}_{\mathsf{s}}$ is undecidable. We present the idea of this new encoding by showing that it at least faithfully encodes the machine $\mathtt{M}_{\mathrm{even}}$. As we will see, what makes it work is that the new encoding is such that, even if they were available, the ambient instructions would not contribute

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

to any more 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 $\mathtt{M}_K$, for an appropriate integer K, as a modification of $\mathtt{M}_{\text{even}}$ that works at an exponential scale (with base K) compared to that of $\mathtt{M}_{\text{even}}$. In particular, $\mathtt{M}_K$ manages to replace the decrement instructions $p_0 : q_0\mathtt{r}_1 \leq q_1$ and $p_1 : q_1\mathtt{r}_1 \leq q_0$ by *programs* (sets of instructions) $\mathcal{P}_0$ and $\mathcal{P}_1$, respectively, that divide the contents of register $\mathtt{r}_1$ by the fixed constant K. For example, the general effect for p_0 being $q_0\mathtt{r}_1^m \leq^{p_0} q_1\mathtt{r}_1^n$ iff $m = n + 1$ is mirrored by $\mathcal{P}_0$ in the sense that $q_0\mathtt{r}_1^M \leq_{\mathcal{P}_0} q_1\mathtt{r}_1^N$ iff $M = K \cdot N$, and consequently $q_0\mathtt{r}_1^{K^{n+1}} \leq_{\mathcal{P}_0} q_1\mathtt{r}_1^{K^n}$; therefore computations in $\mathtt{M}_{\text{even}}$ are simulated in $\mathtt{M}_K$ by storing the contents of $\mathtt{r}_1$ by K^n instead of n. In this case, we will say a term is accepted if it computes a join of configurations of the form $q_f\mathtt{r}_1^{K^0}$ (i.e., $q_f\mathtt{r}_1$), so $q_0\mathtt{r}_1^n \in \mathrm{Acc}(\mathtt{M}_K)$ iff $n = K^{2m}$ for some $m \geq 0$. Thereupon an additional necessary condition for acceptance in $\mathtt{M}_K$ is demanded for configurations labeled by a state $q \in \mathtt{Q}_{\text{even}}$ (independently from the conditions of acceptance in $\mathtt{M}_{\text{even}}$) namely that if $q\mathtt{r}_1^N$ is accepted in $\mathtt{M}_K$ then N must be a power of K.[2] For the equation s, if we choose $K > 2$ it is easily verified that if s is applied

$$q\mathtt{r}_1^n\mathtt{r}_1^m \leq^{\mathsf{s}} q\mathtt{r}_1^n(\mathtt{r}_1^{2m} \vee \mathtt{r}_1^{4m}) = q\mathtt{r}_1^{n+2m} \vee q\mathtt{r}_1^{n+4m},$$

the only way $n + 2m$ and $n + 4m$ are both powers of K is if $m = 0$.[3] In such an instance, the configuration on the left-hand side of the equation appears as a joinand on the right-hand side. Consequently we see that, with respect to being accepted, instances of $\leq^{\mathsf{s}}$ in a computation are superfluous, and we obtain

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

thus $\mathrm{Acc}(\mathtt{M}_K) = \mathrm{Acc}(\mathsf{s}\mathtt{M}_K)$. So, the equation s is *admissible* in the machine $\mathtt{M}_K$.

The reason why this works is that the effect of the inequality s, even when applied repeatedly, is to modify the register values in a linear or polynomial way, but when these values are encoded on an exponential scale the applications of the inequality do not produce modifications on the same scale and thus do not lead to final configurations.

More generally, consider an n-variable simple equation $[\mathrm{D}] : \mathbf{x}_n^{\mathbf{1}} \leq \bigvee_{d\in\mathrm{D}} \mathbf{x}_n^d$. For $[\mathrm{D}]$ to be admissible, and viewing $[\mathrm{D}]$ as an ambient instruction, we need to consider all the substitution instances $\mathbf{t}_n^{\mathbf{1}} \leq \bigvee_{d\in\mathrm{D}} \mathbf{t}_n^d$ of $[\mathrm{D}]$, where the tuple of terms $\mathbf{t}_n = (t_1, \dots, t_n)$ is given by a substitution $\boldsymbol{\sigma} : x_i \mapsto t_i$, for all i. Then for any term s, $s\mathbf{t}_n^{\mathbf{1}} \leq^{\mathrm{D}} s\bigvee_{d\in\mathrm{D}} \mathbf{t}_n^d$ is part of the computation that includes the ambient instructions coming from $[\mathrm{D}]$. It is shown in Lemma 6.5 that if $[\mathrm{D}]$ does not have instances equivalent to a k-mingle equation[4] and $s\bigvee_{d\in\mathrm{D}} \mathbf{t}_n^d$ is accepted in an ACM $\mathtt{M}$, then

---

[2]This definition of acceptance for the machine $\mathtt{M}_K$ is for heuristic convenience. In Section 7, to properly define programs to multiply/divide by K, we will need to add new states and instructions to carry out such computations, as well a fresh variable q_F, acting as a new final state, and a set of instructions that guarantee $q_f\mathtt{r}_1 \leq_{\mathtt{M}_K} q_F$.

[3]Indeed, 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 $K \leq 2$.

[4]Note that if there is an instance of $[\mathrm{D}]$ that is equivalent to *k-mingle* $(x^k \leq x)$ for some $k > 1$, $[\mathrm{D}]$ cannot be admissible for any ACM $\mathtt{M}$: from $q_f^k \leq^{\mathrm{D}} q_f$ we would obtain that q_f^k is accepted, a

the term s contains precisely one state variable and no term t_i contains any state variable. In the case where $\mathtt{M}$ is a 1-ACM, for example $\mathtt{M} = \mathtt{M}_{\text{even}}$, this implies that $s = q\mathtt{r}_1^C$, for some state q and number C, and $\boldsymbol{\sigma}$ is a (1-variable) substitution with $\boldsymbol{\sigma}(x_i) = t_i = \mathtt{r}_1^{\sigma(i)}$, where σ is an n-tuple of natural numbers. Using the equality relation for $\mathbf{A}_{\mathtt{M}}$, this is equivalently written as

$$q\mathtt{r}_1^{C+\sigma\mathbf{1}} \leq^{\mathrm{D}} \bigvee_{d\in \mathrm{D}} q\mathtt{r}_1^{C+\sigma d}, \tag{3}$$

where $\sigma d = \sigma(1)d(1) + \cdots \sigma(n)d(n)$ and $\sigma\mathbf{1} = \sigma(1) + \cdots \sigma(n)$.

Admissibility of [D] in such a 1-ACM $\mathtt{M}$ is the demand that if the right-hand side of the above inequality is accepted in $\mathtt{M}$ then the left-hand side is also accepted (thus making every instance of [D] superfluous). The most naive and obvious way to ensure this is to ask that the left-hand side already appears as one of the joinands on the right-hand side; that is $\sigma\mathbf{1} = \sigma\bar{d}$ for some $\bar{d} \in \mathrm{D}$, hence rendering the substitution instance of [D] by $\boldsymbol{\sigma}$ trivial. Recall that in the case of the machine $\mathtt{M}_K$ constructed from $\mathtt{M}_{\text{even}}$, if a configuration $q\mathtt{r}_1^N$ is accepted in $\mathtt{M}_K$ then N must be some power of K. So, for [D] to be admissible in $\mathtt{M}_K$ the most obvious condition to require is:

> If the exponents in the right-hand side of [D] produced by a 1-variable substitution are translated powers of K (by the same constant), then the substitution instance is trivial.

In symbolic terms this can be written as

$$\begin{gathered}\text{If for some } \sigma \in \mathbb{N}^n \text{ and } C \in \mathbb{N},\\ \text{every } C + \sigma d \text{ is a power of } K, \text{ where } d \in \mathrm{D},\\ \text{then there exists } \bar{d} \in \mathrm{D} \text{ such that } \sigma\bar{d} = \sigma\mathbf{1},\end{gathered} \tag{$\star K$}$$

in which case we say that [D] satisfies $(\star K)$. We also consider the condition $(\star)$: there exists $K > 1$ such that $(\star K)$ holds.

In the following sections we will make rigorous the notion of admissibility and carefully construct the machines $\mathtt{M}_K$.

5.2. Spineless equations. We will now define a class of simple equations, for which their defining subvarieties will have an undecidable word problem. The class is so vast that it is easier to define its complement. We motivate the definition with the following observation.

Consider the machine $\mathtt{M}_{\text{even}}$ from Example 3.3, and the simple equation $\mathsf{d} : x \leq 1 \vee x^2$. As before, it is easy to see that $q_0\mathtt{r}_1^3 \in \mathrm{Acc}(\mathsf{d}\mathtt{M}_{\text{even}}) \setminus \mathrm{Acc}(\mathtt{M}_{\text{even}})$. However, this behavior cannot be remedied by $\mathtt{M}_K$ for any $K > 1$. For example, let $n = (K^4 - K^2)/2$, then $q_0\mathtt{r}_1^{K^2+n} \notin \mathrm{Acc}(\mathtt{M}_K)$ since $K^2 + n \neq K^{2m}$ for any $m \in \mathbb{N}$. However,

$$q_0\mathtt{r}_1^{K^2+n} = q_0\mathtt{r}_1^{K^2}\mathtt{r}_1^n \leq^{\mathsf{d}} q_0\mathtt{r}_1^{K^2}\mathtt{r}_1^0 \vee q_0\mathtt{r}_1^{K^2}\mathtt{r}_1^{2n} = q_0\mathtt{r}_1^{K^2} \vee q_0\mathtt{r}_1^{K^4} \in \mathrm{Acc}(\mathtt{M}_K).$$

---

contradiction. Actually, then the variety of $\mathsf{CRL} + [\mathrm{D}]$ has a decidable word problem. More generally, we note that k-mingle, as well as contraction, are examples of *knotted equations*: equations of the form $x^k \leq x^l$, where $k \neq l$. It is known [17] that all knotted subvarieties of CRL have decidable universal theories, and therefore so do the subvarieties of CRL axiomatized by any set of simple equations Γ for which $\mathsf{CRL} + \Gamma \models x^k \leq x^l$ by [5].

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

By setting $C = K^2$ and $\sigma = (n) \in \mathbb{N}^1$, this also demonstrates that $(\star K)$ is not satisfied by d for any $K > 1$, i.e., d fails $(\star)$. In fact, we will show in Lemma 8.1 that this failure occurs not just for functions of the form $n \mapsto K^n$ but actually for any function on $\mathbb{N}$ with infinite range (in particular those that are actually computable).

The equation d is an example of a *spinal equation* and as explained above, unfortunately it cannot be handled by our work. In general, a basic equation $[f, \mathrm{V}]$ is called *spinal* if it is of the form:

$$[f, \mathrm{V}] : \underbrace{x_1^{f(1)} \cdots x_k^{f(k)}}_{f} \leq \underbrace{(1 \vee)}_{v_0} \ \underbrace{x_1^{v_1(1)}}_{v_1} \vee \underbrace{x_1^{v_2(1)} x_2^{v_2(2)}}_{v_2} \vee \cdots \vee \underbrace{x_1^{v_k(1)} \cdots x_k^{v_k(k)}}_{v_k},$$

where $f \notin \mathrm{V}$, $v_j(j) \neq 0$, and $v_i(j) = 0$ for each $0 \leq i < j \leq k$, and $(1\vee)$ is meant to signify that 1 may or may not be included in the join. Note that if the column vectors of the set $\mathrm{V} = \{(v_0,)v_1, \dots, v_k\}$ are listed in the above order, V becomes an upper-right triangular matrix whose diagonal contains only positive entries. In Corollary 8.2 we establish that spinal equations falsify the condition $(\star K)$ for every $K > 1$.

Definition 5.2. We say that a basic equation $[f, \mathrm{V}]$ is *spinal* if $f \notin \mathrm{V}$, $v_i(i) \neq 0$, and $v_i(j) = 0$, for all $j > i$, for all v_i in V that are not constantly zero. In this case, we will refer to the set V as a *spine*. We say that a simple equation is *prespinal* if it has a spinal equation as the image under some monoidal substitution.

From now on we will only consider monoidal substitutions (the image of every variable is a monoidal term, i.e., a product of variables) and we will refer to them simply as substitutions.

Note that the only one-variable spinal equations are the knotted inequalities $x^n \leq x^m$ (for which we know that they define subvarieties of CRL with decidable quasiequational theory) and their variants $x^n \leq 1 \vee x^m$, where $n \neq m$, for which decidability results are still open.[5] Also, their equivalent simple equations are prespinal, as verified below in Lemma 5.4. As a consequence of the definition, a simple equation [D] is prespinal if and only if $[\mathrm{D} \cup \{\mathbf{0}\}]$ is prespinal.

From Table 1, we see that (i) and (ii) are spinal. The simple equation (vi) is prespinal via the 1-variable substitution $\boldsymbol{\sigma}$ given by $\boldsymbol{\sigma}(x) = x$, $\boldsymbol{\sigma}(y) = x$, and $\boldsymbol{\sigma}(z) = 1$, i.e., $\mathsf{CRL} + (\mathrm{vi}) \models x^2 \leq x$. On the other hand, no trivial equations are prespinal. The general characterization of whether a simple equation is prespinal will be addressed in Section 8, where it is verified that (iii)–(v) in Table 1 are not prespinal right after Theorem 8.6.

Definition 5.3. A simple equation is called *spineless* if it is not prespinal. A simple equation ε for RL is called spineless if $\varepsilon_{\mathcal{C}}$ is spineless.

To demonstrate the vastness of the collection of spineless equations, we will focus our attention only on 1-variable basic equations below. Note that every one-variable basic equation has the form

$$x^n \leq \bigvee_{p \in P} x^p,$$

[5]The only exception being the case where $n > m = 1$, where equations of this form have the finite model property by Theorem 3.15 in [5].

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

	[D]	D		[D]	D
i.	$x \leq x^2$	$\{2\}$	v.	$xyz \leq x^2y \vee y^2z \vee xz^2$	$\begin{Bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 1 & 2 \end{Bmatrix}$
ii.	$x \leq 1 \vee x^2$	$\{0,2\}$			
iii.	$x \leq x^2 \vee x^4$	$\{2,4\}$			
iv.	$xy \leq 1 \vee x^2y \vee x^3y^2$	$\begin{Bmatrix} 0 & 2 & 3 \\ 0 & 1 & 2 \end{Bmatrix}$	vi.	$xyz \leq yz \vee xz^2$	$\begin{Bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 2 \end{Bmatrix}$

TABLE 1. Some simple equations viewed as sets of column vectors.

for some $n > 0$ and for some finite subset P of $\mathbb{N}$ such that $P \neq \{0\}$; we denote such an equation by $[n, P]$. Also, note that $[n, P]$ is trivial iff $n \in P$.

LEMMA 5.4. *Let $[n, P]$ be a 1-variable basic equation. Then the linearization of $[n, P]$ is a spineless simple equation iff $[n, P]$ is trivial or P contains at least two distinct positive integers.*

PROOF. The simple equation resulting from the linearization over CRL of $[n, P]$ is

$$[\mathrm{D}] : \mathbf{x}_n^{\mathbf{1}} \leq \bigvee \left\{ \mathbf{x}_n^d : d \in \mathbb{N}^n, \sum_{i=1}^{n} d(i) \in P \right\}, \tag{4}$$

and is equivalent over CRL to $[n, P]$.[6]

We prove the contraposition for each direction. For the forward direction, if $P = \{p\}$, where $0 < p \neq n$, then $[n, P]$ is spinal by definition and hence [D] is prespinal by its obvious substitution to $[n, P]$: $x_i \mapsto x$, for all i.

For the reverse direction, suppose that $[n, P]$ is nontrivial and P contains distinct positive numbers $p > q$. Then for each $i \leq n$, the terms x_i^p and x_i^q appear as joinands on the right-hand side of [D]. Let $\boldsymbol{\sigma}$ be a monoidal substitution that is non-trivializing for [D]. Then for some $i \leq n$, $\boldsymbol{\sigma} : x_i \mapsto w$ for some monoid term $w \neq 1$. So both w^p and w^q appear as joinands on the right-hand side of $\boldsymbol{\sigma}$[D]. Since $p > q > 0$, $w^p \neq 1 \neq w^q$, and CRL $\not\models w^p = w^q$, so $\boldsymbol{\sigma}$[D] is not spinal, as w^p and w^q contain the same variables. Since $\boldsymbol{\sigma}$ was arbitrary, it follows that $[n, P]$ is spineless. ⊣

In the following sections we undertake a deep analysis of spineless equations, culminating in Corollary 8.14. To highlight this result, we (re)state it here and we use it right afterwards, in Section 5.3, to obtain results (Theorem 5.9) about the equational theory.

THEOREM 5.5. *Let Γ be a finite set of spineless simple equations, then any variety between CRL + Γ and RL has an undecidable word problem.*

[6] By setting $x := x_1 \vee \cdots x_n$ in $[n, P]$, we obtain $\bigvee\{\mathbf{x}_n^a : \sum_{i=1}^n a(i) = n\} \leq \bigvee\left\{\mathbf{x}_n^d : d \in \mathbb{N}^n, \sum_{i=1}^n d(i) \in P\right\}$. Since $\mathbf{x}_n^{\mathbf{1}} \leq \bigvee\{\mathbf{x}_n^a : \sum_{i=1}^n a(i) = n\}$, we obtain [D]. Conversely, by setting $x_i := x$, for all $i \leq n$, in [D], we obtain $[n, P]$.

5.3. From quasiequations to equations in CRL. In this section we exploit the fact that in certain varieties certain quasiequations are equivalent to equations to show that even their equational theory is undecidable, making use of Theorem 5.5.

The *negative cone* of a residuated lattice $\mathbf{A}$ is the set $A^- = \{a \in A : a \leq 1\}$. We will say that a subvariety $\mathcal{V}$ of 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[7]

$$\begin{array}{rcl} (\exists m \in \mathbb{N})(\exists s_1, \dots, s_m \in S)\ \mathsf{CRL} & \models & \prod_{i=1}^m (1 \wedge s_i) \leq t, \\ \text{if and only if} \quad (\exists k \in \mathbb{N})\ \mathsf{CRL} & \models & (1 \wedge \bigwedge S)^k \leq t. \end{array} \tag{5}$$

If $\mathcal{V} \subseteq$ CRL is a negatively n-potent variety, then we obtain[8]

$$(\exists m \in \mathbb{N})(\exists s_1, \dots, s_m \in S)\ \mathcal{V} \models \prod_{i=1}^m (1 \wedge s_i) \leq t \iff \mathcal{V} \models (1 \wedge \bigwedge S)^n \leq t. \tag{6}$$

We consider the following quasiequation and equation:

$$\xi_S(t) : \mathop{\&}_{s \in S} 1 \leq s \implies 1 \leq t, \qquad \varepsilon_S^n(t) : (1 \wedge \bigwedge S)^n \leq t.$$

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

Lemma 5.6. *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*

$$\mathcal{V} \models \xi_S(t) \iff \mathcal{V} \models \varepsilon_S^n(t).$$

Proof. Let $\mathbf{F}_\mathcal{V}$ be the free algebra for $\mathcal{V}$, and define the congruence $C := \mathrm{Cg}(\{(1 \wedge s, s) : s \in S\})$. We denote the quotient algebra 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.[9] Observe that $\mathcal{V} \models \mathop{\&}_{s \in S} 1 \leq s \Rightarrow 1 \leq t$

$$\begin{array}{lll} \iff & \text{in } \mathbf{F}_\mathcal{V}/C, [1 \wedge t]_C = [1]_C & \\ \iff & \text{in } \mathbf{F}_\mathcal{V}, (1 \wedge t) \in M(\{1 \wedge s : s \in S\}) & [6] \\ \iff & \text{in } \mathbf{F}_\mathcal{V}, (\exists m \in \mathbb{N})(\exists s_1, \dots, s_m \in S)\ \prod_{i=1}^m (1 \wedge s_i) \leq t & [6] \\ \iff & (\exists m \in \mathbb{N})(\exists s_1, \dots, s_m \in S)\ \mathcal{V} \models \prod_{i=1}^m (1 \wedge s_i) \leq t & \\ \iff & \mathcal{V} \models (1 \wedge \bigwedge S)^n \leq t & \text{Equation (6).} \dashv \end{array}$$

For an inequality $p : s \leq t$, define the term $p^\rightarrow := s \rightarrow t$. Let $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P}, q_f)$ be an ACM. Define $\mathtt{P}^\rightarrow := \{p^\rightarrow : p \in \mathtt{P}\}$. Then for $u \in A_\mathtt{M}$, the quasiequation $\mathrm{acc}_\mathtt{M}(u)$

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

[8] The reverse direction follows from (5), 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 $\mathcal{V}$.

[9] See Theorem 3.47 in [6].

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

is equivalent to $\xi_{\mathtt{P}\to}(u \to q_f)$. By Lemma 5.6 and Theorem 4.5, we obtain the following:

THEOREM 5.7. *Let $\mathcal{V}$ be a subvariety of* CRL *containing $\mathbf{W}^+_{\mathtt{M}}$, for some k-ACM* $\mathtt{M}$ *and satisfying $(x \wedge 1)^n \leq (x \wedge 1)^{n+1}$ for some $n \geq 1$. Then deciding membership in the equational theory of $\mathcal{V}$ is at least as hard as deciding membership in* $\mathrm{Acc}(\mathtt{M})$.

We say a simple equation ε is *expansive* if it has, as a substitution instance, an equation of the form

$$x^n \leq \bigvee_{j=1}^{m} x^{n+c_j}, \tag{7}$$

for some $n, m \geq 1$ and $c_1, \dots, c_m \geq 1$. It is easy to verify that if ε is expansive then $\mathsf{CRL} + \varepsilon$ is negatively n-potent. We say a variety is *expansive* if it satisfies an expansive equation. As a consequence of Lemma 5.4, if a simple equation is the equivalent linearization of an expansive basic equation where $m \geq 2$ then it is spineless. By the theorem above and Theorem 4.5 we obtain the following:

COROLLARY 5.8. *Let $\mathcal{V}$ be an expansive subvariety of* CRL *containing $\mathbf{W}^+_{\mathtt{M}}$, for some ACM* $\mathtt{M}$. *Then deciding membership in the equational theory of $\mathcal{V}$ is at least as hard as deciding membership in* $\mathrm{Acc}(\mathtt{M})$.

In particular, we prove the following theorem as a consequence of Theorem 5.5 and the corollary above.

THEOREM 5.9. *If Γ is a finite set spineless simple equation containing an expansive equation then variety* $\mathsf{CRL} + \Gamma$ *has an undecidable equational theory.*

§6. Admissibility. We now begin investigating the required features that a machine should have, in order to achieve the exponential encoding. We begin by formalizing the notion of admissibility and its two natural parts.

Let $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P}, q_f)$ be a k-ACM and [D] a n-variable simple equation. We define the relation $\leq^{\mathrm{D}}$ to be the smallest relation containing

$$\mathbf{t}_n^{\mathbf{1}} \leq \bigvee_{d \in \mathrm{D}} \mathbf{t}_n^{d},$$

for all $\mathbf{t}_n \in ((\mathtt{Q} \cup \mathtt{R}_k)^*)^n$, and closed under the inference rules $[\cdot]$ and $[\vee]$. We define the computation relation $\leq_{\mathrm{D}\mathtt{M}}$ as the smallest compatible preorder generated by $\mathtt{P} \cup \leq^{\mathrm{D}}$, and set $\mathrm{Acc}(\mathrm{D}\mathtt{M}) := \{u \in A_{\mathtt{M}} : \exists \mathtt{u}_f \in \mathtt{Fin}(\mathtt{M}),\ u \leq_{\mathrm{D}\mathtt{M}} \mathtt{u}_f\}$.

The construction of $\leq_{\mathrm{D}\mathtt{M}}$ enjoys an analogue to Lemma 3.1(2), and therefore the following analogue to Lemma 3.4(2):

LEMMA 6.1. *Let* $\mathtt{M}$ *be an ACM and* [D] *be a simple equation. For all $u, v \in A_{\mathtt{M}}$, $u \vee v \in$* $\mathrm{Acc}(\mathrm{D}\mathtt{M})$ *if and only if $u \in$* $\mathrm{Acc}(\mathrm{D}\mathtt{M})$ *and $v \in$* $\mathrm{Acc}(\mathrm{D}\mathtt{M})$.

The frame $\mathbf{W}_{\mathrm{D}\mathtt{M}}$ is defined as $\mathbf{W}_{\mathtt{M}}$, but the nuclear relation is defined with respect to $\mathrm{Acc}(\mathrm{D}\mathtt{M})$ instead of $\mathrm{Acc}(\mathtt{M})$.

LEMMA 6.2. *If* $\mathtt{M}$ *is an ACM and* [D] *a simple equation, then* $\mathbf{W}^+_{\mathrm{D}\mathtt{M}} \in \mathsf{CRL} + [\mathrm{D}]$.

PROOF. Let [D] be an n-variable simple equation where $\mathrm{D} = \{d_1, \dots, d_m\}$. It is enough to show that $\mathbf{W}^+_{\mathrm{DM}} \models [\mathrm{D}]$. By Theorem 5.1, this is equivalent to showing $\mathbf{W}_{\mathrm{DM}} \models (\mathrm{D})$, i.e., for all $s \in W$, $\mathbf{t}_n \in W^n$,

$$\frac{\mathbf{t}_n^{d_1}\ N_{\mathrm{DM}}\ s \quad \cdots \quad \mathbf{t}_n^{d_m}\ N_{\mathrm{DM}}\ s}{\mathbf{t}_n^{\mathbf{1}}\ N_{\mathrm{DM}}\ s}\ (\mathrm{D}).$$

If the antecedent of the implication holds, by the definition of N_{DM} and $\leq_{\mathtt{DM}}$ and by Lemma 6.1 we obtain

$$(\forall d \in \mathrm{D})\ \mathbf{t}_n^d\ N_{\mathrm{DM}}\ s \iff (\forall d \in \mathrm{D})\ s\mathbf{t}_n^d \in \mathrm{Acc}(\mathrm{D}\mathtt{M}) \iff \bigvee_{d \in \mathrm{D}} s\mathbf{t}_n^d \in \mathrm{Acc}(\mathrm{D}\mathtt{M}).$$

Now, by the definition of $\leq_{\mathrm{D}\mathtt{M}}$,

$$s\mathbf{t}_n^{\mathbf{1}} \leq_{\mathrm{DM}} s \bigvee_{d \in \mathrm{D}} \mathbf{t}_n^d = \bigvee_{d \in \mathrm{D}} s\mathbf{t}_n^d,$$

hence $s\mathbf{t}_n^{\mathbf{1}} \in \mathrm{Acc}(\mathrm{D}\mathtt{M})$ and $\mathbf{t}_n^{\mathbf{1}}\ N_{\mathrm{DM}}\ s$. Therefore $\mathbf{W}^+_{\mathrm{DM}} \models [\mathrm{D}]$. ⊣

Since $\leq_{\mathtt{M}} \subseteq \leq_{\mathrm{D}\mathtt{M}}$, it follows that $\mathrm{Acc}(\mathtt{M}) \subseteq \mathrm{Acc}(\mathrm{D}\mathtt{M})$. We say a simple equation [D] is *admissible* in a machine $\mathtt{M}$ if $\mathrm{Acc}(\mathtt{M}) = \mathrm{Acc}(\mathrm{D}\mathtt{M})$. As the only difference between $\mathbf{W}_{\mathtt{M}}$ and $\mathbf{W}_{\mathrm{DM}}$ is $\mathrm{Acc}(\mathtt{M})$ and $\mathrm{Acc}(\mathrm{D}\mathtt{M})$, if [D] is admissible in $\mathtt{M}$ then $\mathbf{W}^+_{\mathtt{M}} = \mathbf{W}^+_{\mathrm{DM}}$. Therefore by Lemma 6.2 we obtain the following lemma:

LEMMA 6.3. *If a simple equation* [D] *is admissible in* $\mathtt{M}$, *then* $\mathbf{W}^+_{\mathtt{M}} \in \mathsf{CRL} + [\mathrm{D}]$.

As we will see, admissibility in $\mathtt{M}$ depends on the machine $\mathtt{M}$ as well as the equation [D]. We define the intermediate notions *register-admissibility* and *state-admissibility* to make this distinction clear.

For a given ACM $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P}, q_f)$ and n-variable simple equation [D], we define $\leq^{\mathrm{DR}}$ to be the smallest relation containing,

$$\mathbf{x}_n^{\mathbf{1}} \leq \bigvee_{d \in \mathrm{D}} \mathbf{x}_n^d,$$

for all $\mathbf{x}_n \in (\mathtt{R}_k^*)^n$, and closed under the inference rules $[\cdot]$ and $[\vee]$. Define the new computation relation $\leq_{\mathrm{DR}\mathtt{M}}$ to be the smallest compatible preorder generated by $\mathtt{P} \cup \leq^{\mathrm{DR}}$, and set $\mathrm{Acc}(\mathrm{DR}\mathtt{M}) := \{u \in A_{\mathtt{M}} : \exists \mathtt{u}_f \in \mathtt{Fin}(\mathtt{M}),\ u \leq_{\mathrm{DR}\mathtt{M}} \mathtt{u}_f\}$. Since the effect of [D] is restricted to only register terms in $\leq^{\mathrm{DR}}$, by the same argument as Lemma 3.4(1), it follows that $\mathrm{Acc}(\mathrm{DR}\mathtt{M}) \subseteq \mathtt{ID}(\mathtt{M})$.

It is clear then that $\mathrm{Acc}(\mathrm{DR}\mathtt{M}) \subseteq \mathrm{Acc}(\mathrm{D}\mathtt{M})$, since $\leq^{\mathrm{DR}}$ is merely a restriction of $\leq^{\mathrm{D}}$. In total, we obtain

$$\mathrm{Acc}(\mathtt{M}) \subseteq \mathrm{Acc}(\mathrm{DR}\mathtt{M}) \subseteq \mathrm{Acc}(\mathrm{D}\mathtt{M}).$$

If $\mathrm{Acc}(\mathtt{M}) = \mathrm{Acc}(\mathrm{DR}\mathtt{M})$, then we say [D] is *register-admissible in* $\mathtt{M}$. If $\mathrm{Acc}(\mathrm{DR}\mathtt{M}) = \mathrm{Acc}(\mathrm{D}\mathtt{M})$, we say [D] is *state-admissible in* $\mathtt{M}$. Hence [D] is admissible in $\mathtt{M}$ iff [D] is both register and state-admissible in $\mathtt{M}$. Due to the property that instructions in an ACM replace a single state-variable of a configuration by precisely one state-variable, we show in Lemma 6.5 that state-admissibility is a property of the equation [D] independent of the machine $\mathtt{M}$.

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

6.1. State-admissibility for spineless equations. We say that a simple equation [D] is *mingly* if there exists a one-variable substitution $\boldsymbol{\sigma}$ such that $\boldsymbol{\sigma}[\mathrm{D}] : x^\lambda \leq \bigvee_{d\in\mathrm{D}} x$ for some $\lambda > 1$. That is, if [D] is an equation in n-variables, $\boldsymbol{\sigma}(\mathbf{x}_n^{\mathbf{1}}) = x^\lambda$ and $\boldsymbol{\sigma}(\mathbf{x}_n^d) = x$, for all $d \in \mathrm{D}$.

Of course, this is equivalent over RL to $x^\lambda \leq x$, but since we do not assume idempotency of $\vee$ in $\mathbf{A}_{\mathtt{M}}$, we write $x^\lambda \leq \bigvee_{d\in\mathrm{D}} x$ so as to be explicit about the implementation of the equation in computations.

By definition, mingly equations are prespinal. Equation (vi) from Table 1 is mingly, using the substitution witnessing it is prespinal, while the remaining equations can easily be verified to be not mingly.[10] Since mingly equations are prespinal by definition, we obtain the following result.

Lemma 6.4. *A spineless equation is non-mingly.*

As we will be dealing only with spineless equations, the equations we will consider are not mingly. The following lemma shows that only mingly equations invalidate state-admissibility and also that state-admissibility is independent of the choice of the machine.

Lemma 6.5. *The following are equivalent for any* $\mathtt{M}$ *and simple equation* [D].

1. [D] *is state-admissible in* $\mathtt{M}$.
2. $\mathrm{Acc}(\mathrm{D}\mathtt{M}) \subseteq \mathtt{ID}(\mathtt{M})$.
3. [D] *is not mingly.*

Proof. Assume $\mathtt{M} = (\mathtt{R}_k, \mathtt{Q}, \mathtt{P}, q_f)$ and let [D] be an n-variable simple equation.

$(1 \Rightarrow 2)$ We have that $\mathrm{Acc}(\mathrm{D}\mathtt{M}) = \mathrm{Acc}(\mathrm{D}\mathtt{RM}) \subseteq \mathtt{ID}(\mathtt{M})$.

$(2 \Rightarrow 3)$ Proceeding by contraposition, suppose [D] is mingly. Then $x^\lambda \leq \bigvee_{d\in\mathrm{D}} x$ is a direct substitution image of [D], for $\lambda > 1$. For $x = q_f$, we have

$$q_f^\lambda \leq^{\mathrm{D}} \bigvee_{d\in\mathrm{D}} q_f \in \mathtt{Fin}(\mathtt{M}) \subseteq \mathrm{Acc}(\mathrm{D}\mathtt{M}).$$

Since $\lambda > 1$, it follows that $q_f^\lambda \notin \mathtt{ID}(\mathtt{M})$.

$(3 \Rightarrow 1)$ Proceeding by contraposition, suppose $\mathrm{Acc}(\mathrm{D}\mathtt{RM})$ is a proper subset of $\mathrm{Acc}(\mathrm{D}\mathtt{M})$ and let $t \in \mathrm{Acc}(\mathrm{D}\mathtt{M}) \setminus \mathrm{Acc}(\mathrm{D}\mathtt{RM})$ be a witness with minimal computation

$$t = u_0 \leq^{p_1} u_1 \leq^{p_2} \cdots \leq^{p_N} u_N = \mathtt{u}_f \in \mathtt{Fin}(\mathtt{M}),$$

for some $u_0, u_1, \dots, u_N \in A_{\mathtt{M}}$ and $p_1, \dots, p_N \in \mathtt{P} \cup \{\mathrm{D}\}$. Since $\mathtt{Fin}(\mathtt{M}) \subseteq \mathrm{Acc}(\mathrm{D}\mathtt{RM})$, we have that $t \notin \mathtt{Fin}(\mathtt{M})$ and so $N > 1$. By Lemma 6.1, we may assume $t \in (\mathtt{Q} \cup \mathtt{R}_k)^*$. Since N is minimal, it follows that $p_1 = \mathrm{D}$ and $u_1 \in \mathrm{Acc}(\mathrm{D}\mathtt{RM}) \subseteq \mathtt{ID}(\mathtt{M})$. So,

$$t = s\mathbf{t}_n^{\mathbf{1}} \leq^{\mathrm{D}} \bigvee_{d\in\mathrm{D}} s\mathbf{t}_n^d = u_1,$$

where $s \in (\mathtt{Q} \cup \mathtt{R}_k)^*$ and $\mathbf{t}_n \in ((\mathtt{Q} \cup \mathtt{R}_k)^*)^n$; here we used the fact that the rule $[\vee]$ is not applicable, as $t \in (\mathtt{Q} \cup \mathtt{R}_k)^*$.

[10] Note that the remaining equations are such that each variable appears with degree at least 2 on the right-hand side, so any 1-variable non-trivializing substitution instance will result in a joinand of degree at least 2.

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

Since $u_1 \in \mathtt{ID}(\mathtt{M})$, it follows that $s\mathbf{t}_n^d \in \mathtt{Conf}(\mathtt{M})$ for all $d \in \mathrm{D}$. Also, because $p_1 \neq \mathrm{D}\mathtt{R}$, there is some t_i that contains at least one state variable. As that t_i must also appear on the right-hand side and $s\mathbf{t}_n^{\mathbf{1}} \in \mathtt{Conf}(\mathtt{M})$, it follows that s cannot contain any state variable, so $s \in \mathtt{R}_k^*$. Consequently, every joinand in the right-hand side has a unique t_i containing a state variable.

Therefore, applying the substitution $\boldsymbol{\sigma}$ defined by: $\boldsymbol{\sigma}(x_i) = x$ if t_i contains a state variable and $\boldsymbol{\sigma}(x_i) = 1$ otherwise, yields $x^\lambda \leq x$, where λ is the number of t_i's containing a state variable.

We will show that $\lambda > 1$. If, by way of contradiction, there was a unique t_j containing a state variable, then it would have the form $t_j = qx$ for some $q \in \mathtt{Q}$ and $x \in \mathtt{R}_k^*$ (with $t_i \in \mathtt{R}_k^*$ for all other t_i's) and t_j would appear on all the joinands on the right-hand side. So, we have

$$t = sqx \prod_{i \neq j} t_i \leq^{\mathrm{DR}} \bigvee_{d \in \mathrm{D}} sqx \prod_{i \neq j}^{n} {t_i}^{d(i)} = u_1 \in \mathrm{Acc}(\mathrm{D}\mathtt{RM}),$$

and thus $t \in \mathrm{Acc}(\mathrm{D}\mathtt{RM})$, a contradiction. ⊣

Therefore, in our search for an appropriate machine for spineless equations, state-admissibility will be automatic and will not restrict the type of possible machines.

§7. The exponential encoding. Given a 2-ACM $\mathtt{M} = (\mathtt{R}_2, \mathtt{Q}, \mathtt{P}, q_f)$ (we will later choose as $\mathtt{M}$ a machine with undecidable halting problem) and simple equation [D], our ultimate goal is to construct a new machine $\mathtt{M}'$ "simulating" the machine $\mathtt{M}$ such that $\mathrm{Acc}(\mathrm{D}\mathtt{M}') = \mathrm{Acc}(\mathtt{M}')$. More specifically, for any spineless equation [D] we can construct a 3-ACM $\mathtt{M}_K = (\mathtt{R}_3, \mathtt{Q}_K, \mathtt{P}_K, q_F)$ for which it will be register-admissible, for some $K > 1$ provided by Theorem 8.10, that will simulate the behavior of the 2-ACM $\mathtt{M}$ in the following way:

$$q\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2} \in \mathrm{Acc}(\mathtt{M}) \text{ if and only if } q\mathtt{r}_1^{K^{n_1}}\mathtt{r}_2^{K^{n_2}} \in \mathrm{Acc}(\mathtt{M}_K).$$

So, the content n of a register $\mathtt{r}$ is not represented by $\mathtt{r}^n$ but by $\mathtt{r}^{K^n}$. Computationally, this will be achieved by replacing each increment-$\mathtt{r}$ (decrement-$\mathtt{r}$) instruction by a distinct set of instructions (i.e., a *program*) that will carry out the process of multiplying (dividing) the contents in register $\mathtt{r}$ by K. The auxiliary register $\mathtt{r}_3$ will be necessary to carry out such computations, and the faithfulness of this encoding will be guaranteed by the insistence that accepted configurations are those which have a computation resulting in a finite join of configurations labeled by state q_F where all the registers are empty.

We will first show how we can achieve multiplying the contents of a register $\mathtt{r} \in \mathtt{R}_2$ by a fixed constant $K > 1$. Let $\mathtt{Q}_{+K} = \{a_0, \dots, a_K\}$ be a set of $(K+1)$-many fresh state-variables. We can add K tokens to the contents of the auxiliary register $\mathtt{r}_3$ by starting from the state a_0 and applying the instructions $\mathtt{P}_{+K} = \{+_i : i = 1, \dots, K\}$, where $+_i : a_{i-1} \leq a_i\mathtt{r}_3$, and reaching state a_K. Then for each $0 \leq i < K$, we have

$$a_0 \leq_{+K} a_i\mathtt{r}_3^N \iff N = i,$$

where $\leq_{+K}$ is the computation relation defined in the usual way from $\mathtt{P}_{+K}$. Hence $a_0 x \leq_{+K} a_K y$ iff $y = x\mathtt{r}_3^K$, for each $x, y \in \mathtt{R}_3^*$.

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

Now, if we have N tokens in register $\mathtt{r}$ and we are at state a_K, then by removing one $\mathtt{r}$-token, moving to state a_0 by the instruction $\times_{\text{loop}} : a_K\mathtt{r} \leq a_0$, and using $\mathtt{P}_{+K}$ to add K $\mathtt{r}_3$-tokens and repeating this process, we can essentially exchange N $\mathtt{r}$-tokens for NK $\mathtt{r}_3$-tokens. For example,

$$a_K\mathtt{r}^N \leq^{\times_{\text{loop}}} a_0\mathtt{r}^{N-1} \leq_{+K} a_K\mathtt{r}^{N-1}\mathtt{r}_3^K \leq^{\times_{\text{loop}}} a_0\mathtt{r}^{N-2}\mathtt{r}_3^K \leq_{+K} \cdots a_K\mathtt{r}_3^{NK}.$$

If we set $\leq_{\times\mathtt{r}}$ to be the computation relation defined from $\mathtt{P}_{\mathtt{r}\times K} = \mathtt{P}_{+K} \cup \{\times_{\text{loop}}\}$, then it is easily verified by induction that for each $N, M, n, m \in \mathbb{N}$ and $0 \leq i \leq K$:

$$\begin{array}{lcl} a_K\mathtt{r}^N \leq_{\times\mathtt{r}} a_i\mathtt{r}^n\mathtt{r}_3^m & \iff & KN = Kn + m + (K - i), \\ a_i\mathtt{r}^n\mathtt{r}_3^m \leq_{\times\mathtt{r}} a_K\mathtt{r}_3^M & \iff & M = Kn + m + (K - i). \end{array} \tag{8}$$

Observe that multiplying the contents of register $\mathtt{r}$ is achieved by an iterative process of adding K-many tokens to $\mathtt{r}_3$ and then looping the process by removing one from $\mathtt{r}$. We can define a division program analogously. However, in both cases, we start with tokens in $\mathtt{r}$ and compute the product (or quotient) by K in the register $\mathtt{r}_3$ by emptying $\mathtt{r}$. We would then like our machine to *transfer* those contents back to the original register $\mathtt{r}$ to complete this program.

Let $\mathtt{T_r}$ be the program with fresh states $\mathtt{Q}_{\mathtt{T_r}} = \{t_0, t_1\}$ and instructions $\mathtt{P}_{\mathtt{T_r}} = \{T_-, T_+\}$ given by $T_- : t_0\mathtt{r}_3 \leq t_1$ and $T_+ : t_1 \leq t_0\mathtt{r}$. Defining its computation relation to be $\leq_{\mathtt{T_r}}$, we easily obtain for all $N, M, n, m \in \mathbb{N}$:

$$\begin{array}{lcl} t_0\mathtt{r}_3^N \leq_{\mathtt{T_r}} t_\delta\mathtt{r}^n\mathtt{r}_3^m & \iff & N = n + m + \delta, \\ t_\delta\mathtt{r}^n\mathtt{r}_3^m \leq_{\mathtt{T_r}} t_0\mathtt{r}^M & \iff & n + m + \delta = M. \end{array} \tag{9}$$

If we wish to implement the transfer program after, say, executing the program for multiplying by K, then we need an instruction to switch from the state a_0 to the state t_0. This can naively be achieved by a forking instruction, say $p : a_K \leq t_0$,[11] which would allow for the computation

$$a_K\mathtt{r}^N \leq_{\times\mathtt{r}} a_K\mathtt{r}_3^{NK} \leq^p t_0\mathtt{r}_3^{NK} \leq_{\mathtt{T_r}} t_0\mathtt{r}^{NK}.$$

However, since the instruction p can be applied to any configuration labeled by a_K, even those for which the register $\mathtt{r}$ is nonempty, we also get unwanted instances of the form $a_K\mathtt{r}^N \leq t_0\mathtt{r}^M$ where $M = Km + N - m$ for each $0 \leq m \leq N$. Since we want our simulation to be faithful, we need a way of switching to the transfer program *only when* the register $\mathtt{r}$ is empty.

7.1. The zero-test program. What we are asking for is similar to (what is commonly called) a zero-test instruction of a standard counter machine, i.e., an instruction which is applicable only when a specified register is empty. Since we require our computation relations to be compatible with multiplication, such an insistence is impossible. That is, we insist $q \leq q'$ to entail $qx \leq q'x$ for any term x.

However, following the ideas in [11], we construct a program that has a similar behavior utilizing the insistence that accepted configurations are those that compute final ID's, i.e., finite joins of the configuration labeled by a final state q_F with all registers empty.

[11] Technically, $p : a_0 \leq t_0 \vee t_0$ since $\vee$ is not idempotent, but this technicality is unnecessary for the example.

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

We define the (sub-)machine $ø = (\mathtt{R}_3, \mathtt{Q}_ø, \mathtt{P}_ø, q_F)$, with a fresh set of variables $\mathtt{Q}_ø = \{z_1, z_2, z_3, q_F\}$ and instructions $\mathtt{P}_ø$ are given by:

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

for each $i, j \in \{1, 2, 3\}$ with $i \neq j$, resulting in a total of $6 + 3 = 9$ instructions.

The addition of the auxiliary states z_1, z_2, z_3 is explained by the fact that the role of z_i is to empty the contents of all registers other than $\mathtt{r}_i$ and transition to the final state q_F. Thus it detects situations where $\mathtt{r}_i$ is not already empty, as then it cannot reach a final ID. So, assuming we want to move from state q_{in} to q_{out} only when the register $\mathtt{r}_i$ is empty, we can start a parallel computation, the main branch of which moves to state q_{out} (even if $\mathtt{r}_i$ is non-empty) but the auxiliary branch involving the z_i terminates successfully only if $\mathtt{r}_i$ is empty. This z_i-branch ensures/safeguards that the combined computation acts as intended.

We call the above machine the *zero-test program*, and we denote its computation relation by $\leq_ø$. 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 $\mathtt{M}_K$ will consist of only 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 the program $ø$ 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 $q_{\text{in}}\mathtt{r}_1\mathtt{r}_2\mathtt{r}_3$:

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

As we see, the above (maximal in $ø$) 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, $z_1\mathtt{r}_1\mathtt{r}_2\mathtt{r}_3 \notin \mathrm{Acc}(ø)$ 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:

LEMMA 7.1. $z_i\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \in \mathrm{Acc}(ø)$ *if and only if* $n_i = 0$.

Let $\mathcal{P}$ be a program (i.e., a sub-machine) and $\leq_\mathcal{P}$ be its corresponding computation relation. We define the relation $\sqsubseteq_\mathcal{P}$ on $\mathtt{Conf}(\mathcal{P})$ via $\mathtt{C} \sqsubseteq_\mathcal{P} \mathtt{C}'$ iff $\mathtt{C} \leq_\mathcal{P} \mathtt{C}' \vee u$, where either $u = \bot$ or $u \in \mathrm{ID}(ø)$ with $u \in \mathrm{Acc}(ø)$.[12] If $\mathcal{P}$ contains no $\mathtt{Q}_ø$-instructions (i.e., no instruction $z_i x \leq \cdots$), then $\mathtt{C} \sqsubseteq_\mathcal{P} \mathtt{C}'$ iff there is a computation from $\mathtt{C}$ to $\mathtt{C}' \vee u$ with instructions from $\mathcal{P}$ such that every zero-test was properly applied. Note that $\sqsubseteq_\mathcal{P}$ is transitive on configurations and $\mathtt{C} \leq_\mathcal{P} \mathtt{C}'$ implies $\mathtt{C} \sqsubseteq_\mathcal{P} \mathtt{C}'$. We obtain the following lemma.

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

LEMMA 7.2. *Let p be the instruction $q_{\text{in}} \le q_{\text{out}} \vee z_i$ with distinct $q_{\text{in}}, q_{\text{out}} \notin \mathtt{Q}_{\varnothing}$. For $x, x' \in \mathtt{R}_3^*$, $q_{\text{in}}x \sqsubseteq_{\{p\}} q_{\text{out}}x'$ 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 $q_{\text{in}}x$ is p, so from $q_{\text{in}} \le q_{\text{out}} \vee z_i$ we obtain $q_{\text{in}}x \le^p q_{\text{out}}x \vee z_i x$. Since the only instructions applicable are those from $\{p\}$ and $q_{\text{in}} \ne q_{\text{out}}$, the computation cannot proceed from this configuration. Hence by Lemma 7.1,

$$q_{\text{in}}x \sqsubseteq_{\{p\}} q_{\text{out}}x' \iff x = x' \text{ and } z_i x \le_{\varnothing} q_F \iff x = x' \text{ and } n_i = 0. \qquad \dashv$$

7.2. Multiplying and dividing. We are now ready to faithfully simulate an increment-$\mathtt{r}$ instruction by a program that multiplies the contents of register $\mathtt{r} \in \{\mathtt{r}_1, \mathtt{r}_2\}$ by the fixed constant K. For $p : q_{\text{in}} \le q_{\text{out}}\mathtt{r}$, we define the program $\times(p)$ to have states $\mathtt{Q}_{\times(p)} = \mathtt{Q}_{+K} \cup \mathtt{Q}_{\mathtt{T}_\mathtt{r}}$ and instructions $\mathtt{P}_{\times(p)} = \mathtt{P}_{+K} \cup \mathtt{P}_{\mathtt{T}_\mathtt{r}} \cup \{\times_{\text{in}}, \times_{\mathtt{T}}, \times_{\text{out}}\}$, where

$$\begin{array}{rcrcl} \times_{\text{in}} & : & q_{\text{in}} & \le & a_K \vee z_3, \\ \times_{\mathtt{T}} & : & a_K & \le & t_0 \vee z_i, \\ \times_{\text{out}} & : & t_0 & \le & q_{\text{out}} \vee z_3. \end{array}$$

The instruction $\times_{\text{in}}$ is intended to verify that the auxiliary register $\mathtt{r}_3$ is in fact empty, and initiate the process of storing in $\mathtt{r}_3$ K-times the contents in the active register $\mathtt{r}$. The instruction $\times_{\mathtt{T}}$ is meant to check that all the contents of the active register $\mathtt{r}$ have been emptied (and thus K-times that amount is in $\mathtt{r}_3$), and initiate the transfer program. The instruction $\times_{\text{out}}$ is intended to end the program by transitioning to the state q_{out} only when the transfer is complete, i.e., when $\mathtt{r}_3$ has been emptied. Below is an example of $\times(q_{\text{in}} \le q_{\text{out}}\mathtt{r}_1)$ running on the configuration $q_{\text{in}}\mathtt{r}_1^2\mathtt{r}_2$:

$$\begin{array}{rl} q_{\text{in}}\mathtt{r}_1^2\mathtt{r}_2 & \sqsubseteq_{\{\times_{\text{in}}\}} \quad a_K\mathtt{r}_1^2\mathtt{r}_2 \\ & \le_{\times_{\mathtt{r}_1}} \quad a_K\mathtt{r}_2\mathtt{r}_3^{2K} \\ & \sqsubseteq_{\{\times_{\mathtt{T}}\}} \quad t_0\mathtt{r}_2\mathtt{r}_3^{2K} \\ & \le_{\mathtt{T}_{\mathtt{r}_1}} \quad t_0\mathtt{r}_1^{2K}\mathtt{r}_2 \\ & \sqsubseteq_{\{\times_{\text{out}}\}} \quad q_{\text{out}}\mathtt{r}_1^{2K}\mathtt{r}_2. \end{array}$$

In this way, we obtain the following technical lemma by induction as a consequence of Lemma 7.2 and Equations (8) and (9). We state the lemma for $\mathtt{r} = \mathtt{r}_1$, but the same holds when swapping the roles of $\mathtt{r}_1$ and $\mathtt{r}_2$.

LEMMA 7.3. *Let $p : q_{\text{in}} \le q_{\text{out}}\mathtt{r}_1$ be an increment-$\mathtt{r}_1$ instruction, where $q_{\text{in}}, q_{\text{out}} \notin \mathtt{Q}_{\times(p)}$. Then*

$$q_{in}\mathtt{r}_1^{N_1}\mathtt{r}_2^{N_2}\mathtt{r}_3^{N_3} \sqsubseteq_{\times(p)} q_{out}\mathtt{r}_1^{M_1}\mathtt{r}_2^{M_2}\mathtt{r}_3^{M_3},$$

if and only if $M_1 = KN_1$, $M_2 = N_2$, and $M_3 = N_3 = 0$. In fact, for each $n_1, n_2, n_3 \in \mathbb{N}$ and state $q \in \mathtt{Q}_{\times(p)}$,

1. *$q_{\text{in}}\mathtt{r}_1^{N_1}\mathtt{r}_2^{N_2}\mathtt{r}_3^{N_3} \sqsubseteq_{\times(p)} q\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3}$ iff $N_3 = 0$, $N_2 = n_2$, and*

$$KN_1 = \begin{cases} n_1 + n_3 + \delta & \text{if } q = t_\delta \text{ where } \delta \in \{0, 1\}, \\ Kn_1 + n_3 + (K - \delta) & \text{if } q = a_\delta \text{ where } 0 \le \delta \le K. \end{cases}$$

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

2. $q\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\times(p)} q_{\mathrm{out}}\mathtt{r}_1^{M_1}\mathtt{r}_2^{M_2}\mathtt{r}_3^{M_3}$ *iff* $M_3 = 0$, $M_2 = n_2$, *and*

$$M_1 = \begin{cases} n_1 + n_3 + \delta & \text{if } q = t_\delta \text{ where } \delta \in \{0,1\}, \\ Kn_1 + n_3 + (K - \delta) & \text{if } q = a_\delta \text{ where } 0 \leq \delta \leq K. \end{cases}$$

For a configuration $\mathtt{C} = q\mathtt{r}_1^n\mathtt{r}_2^m$ in $\mathtt{M}$, by $\mathtt{C}_K$ we denote the configuration $q\mathtt{r}_1^{K^n}\mathtt{r}_2^{K^m}$ in $\mathtt{M}_K$.

Corollary 7.4. *Let p be an increment instruction from some 2-ACM $\mathtt{M}$. Then* $\mathtt{C} \leq^p \mathtt{C}'$ *if and only if* $\mathtt{C}_K \sqsubseteq_{\times(p)} \mathtt{C}'_K$ *for any configurations* $\mathtt{C}, \mathtt{C}'$ *in* $\mathtt{Conf}(\mathtt{M})$.

In a completely analogous way, given a decrement-$\mathtt{r}$ instruction $p : q_{\mathrm{in}}\mathtt{r} \leq q_{\mathrm{out}}$, we define the *division by K* program $\div(p)$ as follows. For its set of states $\mathtt{Q}_{\div(p)}$, we define a fresh set of states $\mathtt{Q}_{\mathtt{r}-K} = \{s_0, \dots, s_K\}$ and set $\mathtt{Q}_{\div(p)} = \mathtt{Q}_{\mathtt{r}-K} \cup \mathtt{Q}_{\mathtt{T}_\mathtt{r}}$. We take as its instructions $\mathtt{P}_{\div(p)} = \mathtt{P}_{\mathtt{r}-K} \cup \mathtt{P}_{\mathtt{T}_\mathtt{r}} \cup \{\div_{\mathrm{in}}, \div_{\mathtt{T}}, \div_{\mathrm{out}}\}$, where $\mathtt{P}_{\mathtt{r}-K}$ contains the instruction $\div_{\mathrm{loop}} : s_K \leq s_0\mathtt{r}_3$ and K-many instructions of the form $-_i$: $s_{i-1}\mathtt{r} \leq s_i$, for $1 \leq i \leq K$, and finally

$$\begin{array}{rcrcl} \div_{\mathrm{in}} & : & q_{\mathrm{in}} & \leq & s_0 \vee z_3, \\ \div_{\mathtt{T}} & : & s_0 & \leq & t_0 \vee z_i, \\ \div_{\mathrm{out}} & : & t_0 & \leq & q_{\mathrm{out}} \vee z_3. \end{array}$$

The instruction $\div_{\mathrm{in}}$ is intended to verify that the auxiliary register $\mathtt{r}_3$ is in fact empty, and initiate the process of storing in $\mathtt{r}_3$ the quotient by K of the contents in the active register $\mathtt{r}$. The instruction $\times_{\mathtt{T}}$ is meant to check that all the contents of the active register $\mathtt{r}$ have been emptied (and thus K-divided by that amount is in $\mathtt{r}_3$), and initiate the transfer program. The instruction $\div_{\mathrm{out}}$ is intended to end the program transitioning to the state q_{out} only when the transfer is complete, i.e., when $\mathtt{r}_3$ has been emptied. Below is an example of $\div(q_{\mathrm{in}}\mathtt{r}_1 \leq q_{\mathrm{out}})$ running on the configuration $q_{\mathrm{in}}\mathtt{r}_1^{2K}\mathtt{r}_2$:

$$\begin{array}{rll} q_{\mathrm{in}}\mathtt{r}_1^{2K}\mathtt{r}_2 & \sqsubseteq_{\{\div_{\mathrm{in}}\}} & s_0\mathtt{r}_1^{2K}\mathtt{r}_2 \\ & \leq_{\mathtt{r}_1-K} & s_K\mathtt{r}_1^{K}\mathtt{r}_2 \\ & \leq^{\div_{\mathrm{loop}}} & s_0\mathtt{r}_1^{K}\mathtt{r}_2\mathtt{r}_3 \\ & \leq_{\mathtt{r}_1-K} & s_K\mathtt{r}_2\mathtt{r}_3 \\ & \leq^{\div_{\mathrm{loop}}} & s_0\mathtt{r}_2\mathtt{r}_3^2 \\ & \sqsubseteq_{\{\div_{\mathtt{T}}\}} & t_0\mathtt{r}_2\mathtt{r}_3^2 \\ & \leq_{\mathtt{T}_{\mathtt{r}_1}} & t_0\mathtt{r}_1^2\mathtt{r}_2 \\ & \sqsubseteq_{\{\div_{\mathrm{out}}\}} & q_{\mathrm{out}}\mathtt{r}_1^2\mathtt{r}_2. \end{array}$$

The following technical lemma is easily verified by induction. We state the lemma for $\mathtt{r} = \mathtt{r}_1$, but the same holds by swapping the roles of $\mathtt{r}_1$ and $\mathtt{r}_2$.

Lemma 7.5. *Let $p : q_{\mathrm{in}}\mathtt{r}_1 \leq q_{\mathrm{out}}$ be a decrement-$\mathtt{r}_1$ instruction. Then*

$$q_{in}\mathtt{r}_1^{N_1}\mathtt{r}_2^{N_2}\mathtt{r}_3^{N_3} \sqsubseteq_{\div(p)} q_{out}\mathtt{r}_1^{M_1}\mathtt{r}_2^{M_2}\mathtt{r}_3^{M_3},$$

if and only if $KM_1 = N_1 > 0$, $M_2 = N_2$, and $M_3 = N_3 = 0$. In fact, for each $n_1, n_2, n_3 \in \mathbb{N}$ and state $q \in \mathtt{Q}_{\div(p)}$,

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

1. $q_{\text{in}}\mathtt{r}_1^{N_1}\mathtt{r}_2^{N_2}\mathtt{r}_3^{N_3} \sqsubseteq_{\div(p)} q\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3}$ *iff* $N_3 = 0$, $N_2 = n_2$, *and*
$$N_1 = \begin{cases} n_1 + n_3 + \delta & \textit{if } q = t_\delta \textit{ where } \delta \in \{0,1\}, \\ Kn_1 + n_3 + (K - \delta) & \textit{if } q = s_\delta \textit{ where } 0 \le \delta \le K. \end{cases}$$

2. $q\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \sqsubseteq_{\times(p)} q_{\text{out}}\mathtt{r}_1^{M_1}\mathtt{r}_2^{M_2}\mathtt{r}_3^{M_3}$ *iff* $M_3 = 0$, $M_2 = n_2$, *and*
$$KM_1 = \begin{cases} n_1 + n_3 + \delta & \textit{if } q = t_\delta \textit{ where } \delta \in \{0,1\}, \\ Kn_1 + n_3 + (K - \delta) & \textit{if } q = s_\delta \textit{ where } 0 \le \delta \le K. \end{cases}$$

COROLLARY 7.6. *Let p be a decrement instruction from some 2-ACM $\mathtt{M}$ and $\mathtt{C}, \mathtt{C}'$ be configurations in $\mathtt{M}$. Then $\mathtt{C} \le^p \mathtt{C}'$ if and only if $\mathtt{C}_K \sqsubseteq_{\div(p)} \mathtt{C}'_K$.*

COROLLARY 7.7. *Assume that $p : q_{\text{in}}\mathtt{r} \le q_{\text{out}}$ is a decrement instruction, $\mathtt{C}_{\text{in}}$ is a q_{in}-configuration, $\mathtt{C}$ is a $\mathtt{Q}_{\div(p)}$-configuration, and $\mathtt{C}_{\text{out}}$ is q_{out}-configuration. If $\mathtt{C}_{\text{in}} \sqsubseteq_{\div(p)} \mathtt{C}$ and $\mathtt{C}_{\text{in}} \sqsubseteq_{\div(p)} \mathtt{C}_{\text{out}}$, then $\mathtt{C} \sqsubseteq_{\div(p)} \mathtt{C}_{\text{out}}$.*

PROOF. Since $\mathtt{C}_{\text{in}} \sqsubseteq_{\div(p)} \mathtt{C}$ and $\mathtt{C}_{\text{in}} \sqsubseteq_{\div(p)} \mathtt{C}_{\text{out}}$, by Lemma 7.5 the values of $\mathtt{C}_{\text{in}}$ and $\mathtt{C}$, as well as the values of $\mathtt{C}_{\text{in}}$ and $\mathtt{C}_{\text{out}}$ are linked. Therefore, the values of $\mathtt{C}_{\text{out}}$ and $\mathtt{C}$ are also linked and hence by Lemma 7.5 we obtain $\mathtt{C} \sqsubseteq_{\div(p)} \mathtt{C}_{\text{out}}$. ⊣

7.3. Construction of $\mathtt{M}_K$. Let $\mathtt{M} = (\mathtt{R}_2, \mathtt{Q}, \mathtt{P}, q_f)$ be a 2-ACM and let $K > 1$ be an integer. Since the configuration q_f is accepted in $\mathtt{M}$ by definition, we will need $(q_f)_K = q_f\mathtt{r}_1\mathtt{r}_2$ to be accepted in $\mathtt{M}_K$. To accommodate this, we define the *end program* as follows. For a fresh variable c_F, we define the set of states $\mathtt{Q}_F = \{c_F\}$ and the set of instructions $\mathtt{P}_F = \{F_1, F_2\}$ by:

$$\begin{array}{rcrcl} F_1 & : & q_f\mathtt{r}_1 & \le & c_F, \\ F_2 & : & c_F\mathtt{r}_2 & \le & q_F. \end{array}$$

By $\le_F$ we denote the computation relation for the end program.

LEMMA 7.8. *For $q \in \mathtt{Q}_F \cup \{q_f\}$, $q\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3} \le_F q_F$ if and only if $n_3 = 0$ and*
$$(n_1, n_2) = \begin{cases} (1,1) & \textit{if } q = q_f, \\ (0,1) & \textit{if } q = c_F, \\ (0,0) & \textit{if } q = q_F. \end{cases}$$

We write the instructions $\mathtt{P}$ of $\mathtt{M}$ as the disjoint union $\mathtt{P}_+ \cup \mathtt{P}_- \cup \mathtt{P}_\vee$ of its increment, decrement, and forking instructions, respectively. We can now formally define the 3-ACM simulation of $\mathtt{M}$ to be the machine $\mathtt{M}_K = (\mathtt{R}_3, \mathtt{Q}_K, \mathtt{P}_K, q_F)$, where

- $\mathtt{Q}_K$ is the (disjoint) union of $\mathtt{Q}$, $\mathtt{Q}_\varnothing$, $\mathtt{Q}_F$, $\mathtt{Q}_{\times(p)}$ for each $p \in \mathtt{P}_+$, and $\mathtt{Q}_{\div(p)}$ for each $p \in \mathtt{P}_-$.
- $\mathtt{P}_K$ is the (disjoint) union of $\mathtt{P}_\vee$, $\mathtt{P}_\varnothing$, $\mathtt{P}_F$, $\mathtt{P}_{\times(p)}$ for each $p \in \mathtt{P}_+$, and $\mathtt{P}_{\div(p)}$ for each $p \in \mathtt{P}_-$.
- q_F is the final state of $\mathtt{M}_K$.

Formally, we view all states and instructions in some multiply/divide program $\mathcal{P}(p)$ (where p is an increment/decrement instruction from $\mathtt{M}$) as being labeled by the instruction p, e.g., a state from $\mathtt{Q}_{\mathcal{P}(p)}$ is of the form q^p, and an instruction $\mathtt{P}_{\mathcal{P}(p)}$ is of the form ρ^p. In other words, we make states and instructions in each subprogram disjoint. In fact, since there are no instructions in $\mathcal{P}(p)$ of the form $\cdots \leq q_F \cdots$, we obtain the following useful observation.

LEMMA 7.9. *Let p be an increment or decrement instruction from $\mathtt{M}$ and $\mathcal{P}(p)$ its corresponding program in $\mathtt{M}_K$. If $\mathtt{C}$ is a configuration in $\mathtt{M}_K$ labeled by a state from $\mathtt{Q}_{\mathcal{P}(p)}$ then the only instructions applicable to $\mathtt{C}$ are those from $\mathtt{P}_{\mathcal{P}(p)}$. Furthermore, if q_{out} is the output state of p, then $\mathtt{C}$ being accepted in $\mathtt{M}_K$ implies $\mathtt{C} \sqsubseteq_{\mathcal{P}(p)} \mathtt{C}' \in \mathrm{Acc}(\mathtt{M}_K)$, where $\mathtt{C}'$ is labeled by q_{out}.*

Recall, for a configuration $\mathtt{C} = q\mathtt{r}_1^n\mathtt{r}_2^m$ in $\mathtt{M}$, by $\mathtt{C}_K$ that we denote the configuration $q\mathtt{r}_1^{K^n}\mathtt{r}_2^{K^m}$ in $\mathtt{M}_K$.

LEMMA 7.10. *The following hold for any 2-ACM $\mathtt{M} = (\mathtt{R}_2, \mathtt{Q}, \mathtt{P}, q_f)$ and $K > 1$.*

1. *A configuration $\mathtt{C}$ is accepted in $\mathtt{M}$ iff $\mathtt{C}_K$ is accepted in $\mathtt{M}_K$. Furthermore, any accepted configuration in $\mathtt{M}_K$ labeled by a state from $\mathtt{Q}$ must be of the form $\mathtt{C}_K$ where $\mathtt{C}$ is accepted in $\mathtt{M}$.*
2. *Let p be an increment or decrement instruction of $\mathtt{M}$ and $\mathtt{C}$ a configuration of the corresponding program $\mathcal{P}(p)$ $(\mathcal{P} \in \{\times, \div\})$. Then $\mathtt{C}$ is accepted in $\mathtt{M}_K$ iff there are accepted configurations $\mathtt{C}', \mathtt{C}''$ in $\mathtt{M}$ such that $\mathtt{C}' \leq^p \mathtt{C}''$ and $\mathtt{C}'_K \sqsubseteq_{\mathcal{P}(p)} \mathtt{C} \sqsubseteq_{\mathcal{P}(p)} \mathtt{C}''_K$.*

PROOF. For (1), let $\mathtt{C}$ be a configuration in $\mathtt{M}$. Since there are no q_f-instructions in $\mathtt{M}$ by definition, if $\mathtt{C}$ is labeled by state q_f then it is accepted in $\mathtt{M}$ iff $\mathtt{C} = q_f$, i.e., both registers $\mathtt{r}_1$ and $\mathtt{r}_2$ are empty. By definition, the only q_f-instructions in $\mathtt{M}_K$ are those found in the end program. By Lemma 7.8, the only accepted configuration in $\mathtt{M}_K$ labeled by q_f is $\mathtt{C}_K$. Now, suppose p is a q-instruction from $\mathtt{M}$. Clearly, if p is a forking instruction, then $\mathtt{C} \leq^p \mathtt{C}' \vee \mathtt{C}''$ in $\mathtt{M}$ iff $\mathtt{C}_K \leq^p \mathtt{C}'_K \vee \mathtt{C}''_K$. Otherwise, p is an increment or decrement instruction, and by Corollaries 7.4 and 7.6, $\mathtt{C} \leq^p \mathtt{C}'$ in $\mathtt{M}$ iff $\mathtt{C}_K \sqsubseteq_{\mathcal{P}(p)} \mathtt{C}'_K$ in $\mathtt{M}_K$. The claim therefore follows by induction on the computation lengths.

For (2), consider a configuration $\mathtt{C}$ in $\mathtt{M}_K$ labeled by a state from some program $\mathcal{P}(p)$, where p is an increment or decrement instruction from $\mathtt{M}$. Let q_{in} and q_{out} be the input and output states of p, respectively. By Lemma 7.9, we conclude that if a computation witnesses $\mathtt{C}$ being accepted in $\mathtt{M}_K$ it must implement the output instruction of $\mathcal{P}(p)$. That is $\mathtt{C} \sqsubseteq_{\mathcal{P}(p)} q_{\text{out}}\mathtt{r}_1^{n_1}\mathtt{r}_2^{n_2}\mathtt{r}_3^{n_3}$. By (1), n_1 and n_2 are powers of K while $n_3 = 0$. By Lemmas 7.3 and 7.5, the result follows. ⊣

Let $\tilde{\mathtt{M}}$ be the 2-ACM given by Theorem 3.2. Since membership of $\mathrm{Acc}(\tilde{\mathtt{M}})$ is undecidable, we obtain the following consequence of Lemma 7.10(1):

COROLLARY 7.11. *Membership in the set $\mathrm{Acc}(\tilde{\mathtt{M}}_K)$ is undecidable for $K > 1$.*

7.4. Register-admissibility in $\mathtt{M}_K$. Consider an n-variable simple equation [D], a 2-ACM $\mathtt{M}$, and an integer $K > 1$. To show the register-admissibility of [D] in $\mathtt{M}_K$, we need only show that for each configuration $\mathtt{C}$ in $\mathtt{M}_K$, if the ID $\bigvee_{d \in \mathrm{D}} \mathtt{C}_d$ is obtained by an instance of $\leq^{\mathrm{DR}}$ from $\mathtt{C}$ and $\bigvee_{d \in \mathrm{D}} \mathtt{C}_d$ is accepted in $\mathtt{M}_K$, then $\mathtt{C}$ is accepted in $\mathtt{M}_K$. By Lemma 3.4(2), this implication is equivalently stated as

$$\mathtt{C} \leq^{\mathrm{DR}} \bigvee_{d \in \mathrm{D}} \mathtt{C}_d \quad \& \quad (\forall d \in \mathrm{D})(\mathtt{C}_d \in \mathrm{Acc}(\mathtt{M}_K)) \quad \Longrightarrow \quad \mathtt{C} \in \mathrm{Acc}(\mathtt{M}_K).$$

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

Since we are only considering applications of [D] to the register contents, we can split our analysis into cases depending upon the state $q \in \mathtt{Q}_K$ that labels the configurations. The following useful observation follows from the fact that every variable that appears on the right-hand side of a simple equation appears also on the left-hand side.

LEMMA 7.12. *If a substitution sends all the joinands of a simple equation to* 1, *then it sends all variables of the equation to* 1.

In the following for two tuples σ and d of the same length, σd denotes their dot product. In the next section we will actually view σ as a row-matrix and d as a column-matrix, so σd will be their matrix product. In this way, focusing on the list/column vector d of exponents of the variables in [D] and also on the list/row vector σ of the exponents of the images of the variables via a one-variable substitution, the above lemma can be stated as: for an n-variable simple equation [D], if $\sigma d = 0$ for each $d \in \mathrm{D}$, then σ must be the constantly zero vector $\mathbf{0} \in \mathbb{N}^n$.

As observed in Section 5.1, if $\mathtt{C} \leq^{\mathrm{DR}} \bigvee_{d \in \mathrm{D}} \mathtt{C}_d$ is an instance of $\leq^{\mathrm{DR}}$, we may write $\mathtt{C} = qx\mathbf{x}_n^{\mathbf{1}}$ and $\mathtt{C}_d = qx\mathbf{x}_n^d$ for each $d \in \mathrm{D}$, where $x \in \mathtt{R}_3^*$ and $\mathbf{x}_n = (x_1, \dots, x_n) \in (\mathtt{R}_3^*)^n$. Let $x = \mathtt{r}_1^{C_1}\mathtt{r}_2^{C_2}\mathtt{r}_3^{C_3}$, where $C_1, C_2, C_3 \geq 0$, and for each $j \in \{1, 2, 3\}$, define $\sigma_j \in \mathbb{N}^n$ via $x_i = \mathtt{r}_1^{\sigma_1(i)}\mathtt{r}_2^{\sigma_2(i)}\mathtt{r}_3^{\sigma_3(i)}$, for each $i \in \{1, \dots, n\}$. Then,

$$\mathtt{C} = q\mathtt{r}_1^{C_1+\sigma_1\mathbf{1}}\mathtt{r}_2^{C_2+\sigma_2\mathbf{1}}\mathtt{r}_3^{C_3+\sigma_3\mathbf{1}},$$

and for each $d \in \mathrm{D}$,

$$\mathtt{C}_d = q\mathtt{r}_1^{C_1+\sigma_1 d}\mathtt{r}_2^{C_2+\sigma_2 d}\mathtt{r}_3^{C_3+\sigma_3 d}.$$

LEMMA 7.13. *The zero-test program is register-admissible for any simple equation, and the end program is register-admissible for any non-mingly simple equation.*

PROOF. Let [D] be a simple equation. If q is the final state q_F, then $\mathtt{C}_d$ is accepted iff all registers are empty, i.e., $\mathtt{C}_d = q_F$ for each $d \in \mathrm{D}$. Hence $x = 1$ and $\sigma_j d = 0$ for each $d \in \mathrm{D}$ and $j \in \{1, 2, 3\}$. For each $j \in \{1, 2, 3\}$, this implies that $\sigma_j = \mathbf{0}$, by Lemma 7.12. Therefore $\mathtt{C} = q_F \in \mathrm{Acc}(\mathtt{M}_K)$.

Suppose $q = z_i$, and without loss of generality, let $i = 3$. By Lemma 7.1, $\mathtt{C}_d$ is accepted iff register $\mathtt{r}_3$ is empty, i.e., $\mathtt{C}_d \in \mathrm{Acc}(\varnothing)$ iff $C_3 + \sigma_3 d = 0$, for each $d \in \mathrm{D}$. This implies $C_3 = 0$ and $\sigma_3 d = 0$, for each $d \in \mathrm{D}$. So by Lemma 7.12, $\sigma_3 = \mathbf{0}$. Hence $C_3 + \sigma_3\mathbf{1} = 0$ and $\mathtt{C} \in \mathrm{Acc}(\varnothing) \subseteq \mathrm{Acc}(\mathtt{M}_K)$.

Lastly, suppose $q = c_F$. By Lemma 7.8, $\mathtt{C}_d \in \mathrm{Acc}(F)$ 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 $C_2 + \sigma_2 d = 1$. Again, by Lemma 7.12, $\sigma_1 = \sigma_3 = \mathbf{0}$. Let $\lambda = \sigma_2\mathbf{1}$. Then λ is positive since [D] is a simple equation. If $\lambda = 1$, then $\mathtt{C} = c_F\mathtt{r}_2$ and we are done. If $\lambda \neq 1$ then σ_2 is a substitution witnessing that [D] is mingly. ⊣

Now, suppose $\mathtt{C}$ is labeled by a state $q \in \mathtt{Q}$ from $\mathtt{M}$. By Lemma 7.10, $\mathtt{C}_d$ is accepted in $\mathtt{M}_K$ only if the contents of the registers $\mathtt{r}_1, \mathtt{r}_2$ are each powers of K and the register $\mathtt{r}_3$ is empty. That is, $C_1 + \sigma_1 d$ and $C_2 + \sigma_2 d$ are powers of K and $C_3 + \sigma_3 d = 0$. On the one hand, Lemma 7.12 ensures that σ_3 is the zero vector and $C_3 = 0$, and so $\mathtt{r}_3$ is empty in $\mathtt{C}$.

By the motivation in Section 5.1, a natural condition to consider would be to stipulate that [D] satisfies $(\star K)$. In such a case, if $C_1 + \sigma_1 d$ is a power of K for each $d \in \mathrm{D}$ then there exists $\bar{d} \in \mathrm{D}$ such that $\sigma_1 \mathbf{1} = \sigma_1 \bar{d}$. Similarly, if $C_2 + \sigma_2 d$ is a power of K for each $d \in \mathrm{D}$, then there exists $\bar{d}' \in \mathrm{D}$ such that $\sigma_1 \mathbf{1} = \sigma_1 \bar{d}'$. However, there is no reason *a priori* that entails $\bar{d} = \bar{d}'$ and thus $\mathtt{C} \in \{\mathtt{C}_d : d \in \mathrm{D}\}$, which would be sufficient to ensure that $\mathtt{C}$ would be accepted if $\bigvee_{d \in \mathrm{D}} \mathtt{C}_d$ were accepted.

Since the most naive and obvious way to ensure acceptance is to ask that the left-hand side $\mathtt{C}$ appears as one of the joinands $\mathtt{C}_d$ on the right-hand side, it is sufficient to stipulate that [D] satisfies the following condition:

> If the exponents of each variable in the right-hand side of [D] produced by a 2-variable substitution are translated powers of K, then the substitution instance is trivial.

In symbolic terms this can be written as:

$$\begin{array}{c} \text{For all } \sigma, \sigma' \in \mathbb{N}^n \text{ and for all } C, C' \in \mathbb{N}, \\ \text{if } C + \sigma d \text{ and } C' + \sigma' d \text{ are powers of } K \text{ for each } d \in \mathrm{D}, \\ \text{then there exists } \bar{d} \in \mathrm{D} \text{ such that } \sigma \bar{d} = \sigma \mathbf{1} \text{ and } \sigma' \bar{d} = \sigma' \mathbf{1}. \end{array} \tag{$\star\star K$}$$

In this case, we say [D] satisfies $(\star\star K)$. We also consider the condition $(\star\star)$: there exists $K > 1$ such that $(\star\star K)$ holds. Note that, by setting $\sigma = \sigma'$, we see that if [D] satisfies $(\star\star K)$ then it satisfies $(\star K)$.[13] So, we obtain the following lemma.

LEMMA 7.14. *If a simple equation satisfies* $(\star\star)$ *then it satisfies* $(\star)$.

It is clear then that when $q \in \mathtt{Q}$, if [D] satisfies $(\star\star K)$ then the acceptance of $\bigvee_{d \in \mathrm{D}} \mathtt{C}_d$ in $\mathtt{M}_K$ implies the acceptance of $\mathtt{C}$ in $\mathtt{M}_K$ by Lemma 7.10(1) and the observations above.

As it turns out, the remaining cases can be reduced to the above, and so satisfying the condition $(\star\star K)$ alone is sufficient to ensure register-admissibility. The only remaining cases to verify are when the state q is internal to a multiply or divide by K program. Let $p \in \mathtt{P}$ be some increment or decrement instruction for $\mathtt{M}$. The idea is that if an instance of [D], which leads to acceptance in $\mathtt{M}_K$, occurs internal to a program $\mathcal{P}(p)$ then, by using Lemma 7.10(2) such an instance could have *equivalently* occurred at the end (or beginning) of executing the program $\mathcal{P}(p)$. Without loss of generality, suppose the instruction p acts on register $\mathtt{r}_1$ with input and output states q_{in} and q_{out}, respectively.

For instance, if q is a transfer state $q = t_\delta$, where $\delta \in \{0, 1\}$, then by Lemmas 7.3(2) and 7.5(2),

$$\mathtt{C} \sqsubseteq_{\mathcal{P}(p)} \mathtt{C}' := q_{\text{out}} \mathtt{r}_1^{(C_1 + \sigma_1 \mathbf{1}) + (C_3 + \sigma_3 \mathbf{1}) + \delta} \mathtt{r}_2^{C_2 + \sigma_2 \mathbf{1}},$$

and by Lemmas 7.3(2), 7.5(2), and 7.10(2), for each $d \in \mathrm{D}$,

$$\mathtt{C}_d \sqsubseteq_{\mathcal{P}(p)} \mathtt{C}'_d := q_{\text{out}} \mathtt{r}_1^{(C_1 + \sigma_1 d) + (C_3 + \sigma_3 d) + \delta} \mathtt{r}_2^{C_2 + \sigma_2 d} \in \mathrm{Acc}(\mathtt{M}_K).$$

[13] Surprisingly, we prove in Theorem 8.10 that the converse holds for all K sufficiently large.

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

We see that by setting $C = C_1 + C_3 + \delta$, $C' = C_2$, $\sigma = \sigma_1 + \sigma_3$, and $\sigma' = \sigma_2$, we obtain the instance $\mathtt{C}' \leq^{\mathrm{D}} \bigvee_{d\in\mathrm{D}} \mathtt{C}'_d \in \mathrm{Acc}(\mathtt{M}_K)$. Since $\mathtt{C} \leq_{\mathtt{M}_K} \mathtt{C}'$ and $\mathtt{C}'$ is accepted ($\mathtt{C}'$ is labeled by state $q_{\mathrm{out}} \in \mathtt{Q}$, which was handled above), it follows that $\mathtt{C}$ is accepted.

Similarly, if q is a multiply state $q = a_\delta$, for some $\delta \leq K$, then by Lemma 7.3(2),

$$\mathtt{C} \sqsubseteq_{\times(p)} \mathtt{C}' := q_{\mathrm{out}}\mathtt{r}_1^{K(C_1+\sigma_1\mathbf{1}+K-\delta)+(C_3+\sigma_3\mathbf{1})}\mathtt{r}_2^{C_2+\sigma_2\mathbf{1}},$$

and by Lemmas 7.3(2) and 7.10(2), for each $d \in \mathrm{D}$,

$$\mathtt{C}_d \sqsubseteq_{\times(p)} \mathtt{C}'_d := q_{\mathrm{out}}\mathtt{r}_1^{K(C_1+\sigma_1 d+K-\delta)+(C_3+\sigma_3 d)}\mathtt{r}_2^{C_2+\sigma_2 d} \in \mathrm{Acc}(\mathtt{M}_K).$$

So by setting $C = KC_1 + C_3 + K - \delta$, $C' = C_2$, $\sigma = K\sigma_1 + \sigma_3$, and $\sigma' = \sigma_2$, we obtain the instance $\mathtt{C}' \leq^{\mathrm{D}} \bigvee_{d\in\mathrm{D}} \mathtt{C}'_d \in \mathrm{Acc}(\mathtt{M}_K)$. Since $\mathtt{C} \leq_{\mathtt{M}_K} \mathtt{C}'$ and $\mathtt{C}'$ is accepted ($\mathtt{C}'$ is labeled by state $q_{\mathrm{out}} \in \mathtt{Q}$, which was handled above), it follows that $\mathtt{C}$ is accepted.

Lastly, we consider when q is a division state $q = s_\delta$, for some $\delta \leq K$. By Lemma 7.5(2),

$$\mathtt{C}' := q_{\mathrm{in}}\mathtt{r}_1^{(C_1+\sigma_1\mathbf{1}+\delta)+K(C_3+\sigma_3\mathbf{1})}\mathtt{r}_2^{C_2+\sigma_2\mathbf{1}} \sqsubseteq_{\div(p)} \mathtt{C},$$

and by Lemmas 7.5(2) and 7.10(2), for each $d \in \mathrm{D}$,

$$\mathtt{C}'_d := q_{\mathrm{in}}\mathtt{r}_1^{(C_1+\sigma_1 d+\delta)+K(C_3+\sigma_3 d)}\mathtt{r}_2^{C_2+\sigma_2 d} \sqsubseteq_{\div(p)} \mathtt{C}_d \sqsubseteq_{\div(p)} \mathtt{C}''_d \in \mathrm{Acc}(\mathtt{M}_K),$$

where $\mathtt{C}''_d$ is the unique output configuration of $\div(p)$ labeled by q_{out}.

Now, it is clear that by setting $C = C_1 + KC_3 + \delta$, $C' = C_2$, $\sigma = \sigma_1 + K\sigma_3$, and $\sigma' = \sigma_2$, we have that $\mathtt{C}' \leq^{\mathrm{D}} \bigvee_{d\in\mathrm{D}} \mathtt{C}'_d$. Hence $\mathtt{C}' = \mathtt{C}'_{\bar{d}}$ for some $\bar{d} \in \mathrm{D}$ by ($\star\star K$), and so $\mathtt{C}'_{\bar{d}} \sqsubseteq_{\div(p)} \mathtt{C}$. Since $\mathtt{C}'_{\bar{d}} \sqsubseteq_{\div(p)} \mathtt{C}''_{\bar{d}}$, by Corollary 7.7, it follows that $\mathtt{C} \sqsubseteq_{\div(p)} \mathtt{C}''_{\bar{d}}$. Therefore $\mathtt{C}$ is accepted in $\mathtt{M}_K$ if $\bigvee_{d\in\mathrm{D}} \mathtt{C}_d$ is accepted in $\mathtt{M}_K$.

By the arguments above the following lemma is established:

LEMMA 7.15. *Let* $\mathtt{M}$ *be a* 2*-ACM and* $K > 1$*. If a non-mingly simple equation satisfies* ($\star\star K$) *then it is register-admissible in* $\mathtt{M}_K$.

7.5. Condition ($\star\star$) and undecidability. Assume that [D] is a non-mingly simple equation that satisfies ($\star\star$). Since it is non-mingly, by Lemma 6.5 we get that [D] is state-admissible in any machine. Since it also satisfies ($\star\star$), by Lemma 7.15 we have that [D] is register-admissible in $\mathtt{M}_K$ for some integer $K > 1$, where $\mathtt{M}$ is any machine. In particular, [D] is admissible in $\tilde{\mathtt{M}}_K$, where $\tilde{\mathtt{M}}$ is the machine with undecidable halting problem. By Corollary 7.11, the machine $\tilde{\mathtt{M}}_K$ has an undecidable set of accepted configurations for any $K > 1$. By Lemma 6.3 we obtain $\mathbf{W}^+_{\tilde{\mathtt{M}}_K} \in \mathsf{CRL} + [\mathrm{D}]$. Therefore, $\mathsf{CRL} + [\mathrm{D}]$ has an undecidable word problem by Theorem 4.5. This proves the following result.

COROLLARY 7.16. *For any finite set* Γ *of non-mingly equations that satisfy* ($\star\star$)*, every subvariety of* RL *containing* $\mathsf{CRL} + \Gamma$ *has an undecidable word problem.*

As motivation for the general case, we show that the 1-variable basic equations $[n, P] : x^n \leq \bigvee_{p\in P} x^p$, where P contains at least two distinct positive integers, considered in Lemma 5.4, define varieties with undecidable word problem. The results of the next section will show that this holds for many more equations, all spineless equations.

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

Theorem 7.17. *Let* $[n, P]$ *be a* 1*-variable basic equation where* P *contains at least two distinct positive integers. Then the variety* $\mathsf{CRL} + [n, P]$ *has an undecidable word problem. If additionally* P *only contains integers strictly greater than* n*, then the variety* $\mathsf{CRL} + [n, P]$ *has an undecidable equational theory.*

Proof. Let [D] be the n-variable simple equation that is the linearization of $[n, P]$ over CRL (given by Equation (4) in Lemma 5.4) and let $p, q \in P$ be such that $p > q > 0$. Note that by Lemma 5.4, [D] is spineless and hence it is not mingly by Lemma 6.4. By Corollary 7.16, to establish the first claim it is enough to show [D] satisfies ($\star\star$). We will show that [D] satisfies ($\star\star K$), for every $K > 1 + p - q$; since $p > q$, this implies that $K > 1$.

Assume that there exist $C, C' \in \mathbb{N}$ and 1-variable substitutions σ, σ' such that $C + \sigma d$ and $C + \sigma d'$ are powers of K for each $d \in \mathrm{D}$. We will show that σ and σ' are trivial substitutions (i.e., all entries are 0), and hence $\sigma\mathbf{1} = 0 = \sigma\bar{d}$ and $\sigma'\mathbf{1} = 0 = \sigma'\bar{d}$ for every $\bar{d} \in \mathrm{D}$.

Arguing towards contradiction, suppose that σ is nontrivial with $\sigma(i) > 0$ for some $i \leq n$. Now (by Equation (4)) the terms x_i^p and x_i^q appear as joinands on the right-hand side of [D], i.e., D contains d and d' such that $d(i) = p$, $d'(i) = q$, and $d(j) = d'(j) = 0$ for each $j \neq i$. By the assumption on σ and C, $C + \sigma d = K^{a+b}$ and $C + \sigma d' = K^a$, for some $a, b \in \mathbb{N}$, with $b > 0$ since $p > q$. We have that,

$$K^a(K-1) \leq K^a(K^b - 1) = K^{a+b} - K^a = \sigma d - \sigma d'.$$

Also, $\sigma d = \sigma(i)p$ and $\sigma d' = \sigma(i)q$ (by definition of d, d'), so we obtain

$$\sigma d - \sigma d' = \sigma(i)p - \sigma(i)q = \sigma(i)(p-q) \leq K^a(p-q),$$

where the last inequality follows from $\sigma(i) \leq \sigma(i)q \leq K^a$; note that $q \geq 1$. Combining the inequalities we obtain $K - 1 \leq p - q$ and $K \leq 1 + p - q$, a contradiction. Hence [D] satisfies ($\star\star K$). Furthermore, if all elements from P are larger then n, then $[n, P]$ is an expansive equation. Therefore the second claim follows by Corollary 5.8. ⊣

§8. Characterization of spineless equations. In this section we prove that a simple equation is spineless if and only if it satisfies ($\star\star K$) for every K sufficiently large.

8.1. Basic and simple equations of CRL **as sets of tuples.** Given the natural ordering of the variable set $\{x_i : i \in \mathbb{Z}^+\}$, note that using the above-mentioned vector notation, every commutative monoid term can be written in the form $\mathbf{x}_n^f$, for some $n \in \mathbb{Z}^+$ and some n-tuple f 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 is isomorphic to the above multiplicative rendering by exactly the map $f \mapsto \mathbf{x}_\infty^f$. Up to now we have favored the multiplicative representation due to its connection with machines, but from now on we will use the additive one as it connects better with the linear algebra arguments of this section. 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 $\mathbb{F}^\top$ of the row vectors, which are the transposes of the elements of $\mathbb{F}$. In particular, for $f \in \mathbb{F}$ and $\sigma \in \mathbb{F}^\top$, 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 subset S of $\mathbb{Z}^+$ we define $\mathbb{F}_S$ to be the set of eventually zero functions from S to $\mathbb{N}$, so $\mathbb{F} = \mathbb{F}_{\mathbb{Z}^+}$; we identify $\mathbb{F}_S$ with the corresponding subset of $\mathbb{F}$ in the natural way, as every function in $\mathbb{F}_S$ is the restriction to S of the function in $\mathbb{F}$ that is defined to be zero outside S. We write $\mathbb{F}_n$ for $\mathbb{F}_{\{1,\dots,n\}}$. So, if $f \in \mathbb{F}$ with support included in $\{1, \dots, n\}$, we will identify f with the corresponding element of $\mathbb{F}_n$. We define sets $\mathbb{F}_S^\top$ and $\mathbb{F}_n^\top$ in a similar way. Therefore, $\mathbb{F}_n$ is the set of all $n \times 1$ matrices and $\mathbb{F}_n^\top$ is the set of all $1 \times n$ matrices.

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 each $n \in \mathbb{Z}^+$, we define the column vector $\mathbf{1}_n \in \mathbb{F}$ to contain 1 in its first n entries and 0 everywhere else. 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 subsets A of $\mathbb{F}$ in basic equations $[f, A]$, we may view A as a subset of $\mathbb{F}_n$, where n is the largest index in $\mathrm{supp}(A \cup \{f\})$ and, in this way, will only consider substitutions $\boldsymbol{\sigma} : \mathbb{F}_n \to \mathbb{F}_k$, in which case the associated $M_{\boldsymbol{\sigma}}$ is a $k \times n$ matrix; in this case, we say that $\boldsymbol{\sigma}$ is a k-*variable substitution*. We will write $\sigma_i \in \mathbb{F}_n^\top$ for the i-th row of $M_{\boldsymbol{\sigma}}$ for each $i \le k$ and also $M_{\boldsymbol{\sigma}} = [\sigma_i]_{i=1}^k$. Abusing notation, we will identify $\boldsymbol{\sigma}$ with $M_{\boldsymbol{\sigma}}$ and also we use $\boldsymbol{\sigma}[f, A]$ for the resulting basic inequality. As we have seen in the statement of condition $(\star\star)$, 1-variable substitutions play an important role. Actually, every substitution $\boldsymbol{\sigma}$ is rendered as the product of 1-variable substitutions σ_i (the ones corresponding to the rows of $M_{\boldsymbol{\sigma}} = [\sigma_i]_{i=1}^k$) as for every variable x_j, $\boldsymbol{\sigma}(x_j) = \sigma_1(x_j) \cdot \sigma_2(x_j) \cdots \sigma_k(x_j)$, when using multiplicative notation, and as a sum of 1-variable substitutions σ_i as for every $\mathbf{e}_j$, we have $\boldsymbol{\sigma}(\mathbf{e}_j) = \sigma_1(\mathbf{e}_j) + \sigma_2(\mathbf{e}_j) + \cdots + \sigma_k(\mathbf{e}_j)$, when using additive notation.

8.2. Spinal equations. Let $[f, \mathrm{V}]$ be a k-variable spinal equation. We define $v_0 := \mathbf{0}$ and $\mathrm{V}_\mathbf{0} = \mathrm{V} \cup \{v_0\}$. Using additive notation, it follows from Definition 5.2:

1. V contains a subset V_+ consisting of $k \ge 1$ many vectors $v_1, \dots, v_k$, where $v_j(i)$ is positive if $i = j$ and zero if $i > j$.
2. V is exactly either V_+ or $\mathrm{V}_\mathbf{0}$.
3. f is a vector in $\mathbb{F}_k$ with all entries positive such that $f \notin \mathrm{V}$.

We write $[v_1 \ \cdots \ v_k]$ for the matrix with columns $v_1, \dots, v_k$, in that order. Observe that (1) is equivalent to $[v_1 \ \cdots \ v_k]$ being a $k \times k$ upper-triangular matrix whose diagonal entries are positive.

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

Using this additive perspective, we will demonstrate why spinal equations fail to satisfy the condition $(\star\star)$, and thus the argument for register-admissibility in the machines $\mathtt{M}_K$ found in Lemma 7.15 is not applicable to extensions by such equations. In fact, we prove a much stronger property for spinal equations which entails such an argument will fail, not just for our exponential encoding, but for any similar sort of encoding in general.[14]

To that aim, for a set S of natural numbers, we consider the following property:

> If the exponents in the right-hand side of [D] produced by a 1-variable substitution are in a translation of S (by the same constant), then the substitution instance is trivial.

In symbolic terms this can be written as

$$\begin{gathered}\text{If for some } \sigma \in \mathbb{F}_n^\top \text{ and } C \in \mathbb{N},\\ \text{every } C + \sigma d \text{ is in } S, \text{ for } d \in \mathrm{D},\\ \text{then there exists } \bar{d} \in \mathrm{D} \text{ such that } \sigma\bar{d} = \sigma\mathbf{1}.\end{gathered} \tag{$\star S$}$$

In more compact terms, this can be written as

$$\forall\sigma \in \mathbb{F}_n^\top, \forall C \in \mathbb{N}(C + \sigma\mathrm{D} \subseteq S \Rightarrow \sigma\mathbf{1} \in \sigma\mathrm{D}).$$

Clearly, what we called $(\star K)$ is simply $(\star S)$, where S is the set of all powers of K. In Lemma 8.1, we essentially show that $(\star S)$ fails for any prespinal equation ε and infinite set S.

Lemma 8.1. *If a simple equation satisfies $(\star S)$ for an infinite subset S of $\mathbb{N}$, then it is spineless.*

Proof. We argue by contraposition, assuming that a simple equation ε is prespinal. So there is a substitution $\boldsymbol{\sigma}$ such that $[f, \mathrm{V}] := \boldsymbol{\sigma}\varepsilon$ is a spinal equation, where every column vector of V has k entries/rows. We will construct a 1-variable substitution $\tau = [t_1 \;\; t_2 \;\; \cdots \;\; t_k] \in \mathbb{F}_k^\top$ such that $C + \tau\mathrm{V}_{\mathbf{0}} \subseteq S$, for some C, and $\tau f \notin \tau\mathrm{V}_{\mathbf{0}}$ (hence also $\tau f \notin \tau\mathrm{V}$, as $\mathrm{V} \subseteq \mathrm{V}_{\mathbf{0}}$); let $\mathrm{V}_{\mathbf{0}} = \{v_0, v_1, \dots, v_k\}$. This will imply that ε falsifies $(\star S)$ by the 1-variable substitution $\tau\boldsymbol{\sigma}$ and constant C.

First note that for any $\tau \in \mathbb{F}_k^\top$ we have $\tau v_0 = 0$, so $C + \tau\mathrm{V}_{\mathbf{0}} \subseteq S$ iff $C \in S$ and $C + \tau\mathrm{V}_+ \subseteq S$. Observe that $V_+ := [v_1 \;\; \cdots \;\; v_k]$ is an upper-triangular $k \times k$ matrix whose entries are non-negative integers; note the different font from the set V_+. Furthermore, the determinant $\delta := \det V_+ = v_1(1) \cdots v_k(k)$ is positive since $v_n(n)$ is positive for each $1 \leq n \leq k$ by definition. Hence V_+ is invertible and $V_+^{-1} = \delta^{-1} \operatorname{adj} V_+$, where the adjoint adj V_+ is an upper-triangular matrix with integer entries which furthermore has positive entries on its diagonal (each of the form $\delta/v_n(n)$).

Now, if $C \in S$ and $\tau \in \mathbb{F}_k^\top$ then

$$C + \tau\mathrm{V}_+ \subseteq S \iff \tau V_+ \in (S - C)^k \iff \tau \in (S - C)^k V_+^{-1}.$$

[14]Specifically, we mean the following: Let $\mathtt{M}$ be a 2-ACM and $\phi : \mathbb{N} \to \mathbb{N}$ any (computable) function with infinite range. Let $\mathtt{M}_\phi$ be an ACM constructed so that the register contents $\langle n, m\rangle$ of a configuration from $\mathtt{M}$ are stored as $\langle \phi(n), \phi(m), 0, \dots, 0\rangle$ in $\mathtt{M}_\phi$, and programs constructed so that increments $n \mapsto n + 1$ [decrements $n \mapsto n - 1$] of a register in $\mathtt{M}$ are simulated by $\phi(n) \mapsto \phi(n + 1)$ [$\phi(n) \mapsto \phi(n - 1)$] in $\mathtt{M}_\phi$. Lemma 8.1 ensures that the corresponding argument for register-admissibility is not valid for spinal equations without having more information about Acc($\mathtt{M}$).

Observe that

$$(S-C)^k V_+^{-1} = (S-C)^k \frac{1}{\delta} \operatorname{adj} V_+ = \left(\frac{S-C}{\delta}\right)^k \operatorname{adj} V_+.$$

Therefore,

$$C + \tau \mathrm{V}_+ \subseteq S \iff \tau \in \left(\frac{S-C}{\delta}\right)^k \operatorname{adj} V_+.$$

We claim that there is a $C \in S$ such that the set $(S-C)/\delta$ has an infinite subset in the positive integers. Indeed, since S is infinite there exists a coset $C + \mathbb{N}\delta$ that has infinite intersection with S, where we can take $C \in S$ without loss of generality; let $\bar{\mathbb{N}} \subseteq \mathbb{N}$ be the infinite set such that $C + \bar{\mathbb{N}}\delta$ is the intersection of S with $C + \mathbb{N}\delta$. Hence $\bar{\mathbb{N}}$ is such an infinite subset of $(S-C)/\delta$, and actually $0 \in \bar{\mathbb{N}}$ since $C \in S$. Consequently, if $\tau \in \bar{\mathbb{N}}^k \operatorname{adj} V_+$, then $C + \tau \mathrm{V}_+ \subseteq S$. Note that $\bar{\mathbb{N}}^k \operatorname{adj} V_+$ is an infinite set and all if its entries are integers, while we need $\tau \in \mathbb{F}_k^\top$. Therefore, it is enough to be able to find $[x_1 \cdots x_k] \in \bar{\mathbb{N}}^k$ such that $[t_1\, t_2 \cdots t_k] = [x_1 \cdots x_k] \operatorname{adj} V_+$, where the entries t_i are nonnegative and further $\tau f \notin \tau \mathrm{V}_{\mathbf{0}}$.

Note that since $\operatorname{adj} V_+$ is upper-triangular, the value of t_n, for each $n \leq k$, is determined only by the values $x_1, \dots, x_n$; t_n is a linear combination of only $x_1, \dots, x_n$. This allows us to recursively choose the values of the x_i's, in order to specify the values t_i one-by-one. Furthermore, at the recursive step where we have already determined the values of $x_1, \dots, x_{n-1}$, the value of x_n can be chosen arbitrarily large from the infinite set $\bar{\mathbb{N}}$; moreover, in the linear combination specifying t_n the coefficient of x_n is the (n,n)-entry of $\operatorname{adj} V_+$, which is positive; this allows for the value of t_n to be as large as we want (in particular nonnegative). Therefore, the only thing that we have to ensure is that $x_1, \dots, x_n$ are chosen in $\bar{\mathbb{N}}$ so that furthermore $\tau f \notin \tau \mathrm{V}_{\mathbf{0}}$, i.e., $\tau f \neq \tau v_i$ for all $1 \leq i \leq k$. Below we first prove that $\tau f \neq \tau v_k$ and then that $\tau f > \tau v_i$, for all $i < k$.

Since $[f, \mathrm{V}]$ is a spinal equation, we have $f \notin \mathrm{V}$ and in particular $f \neq v_k$. Let m be the largest number in $\{1, \dots, k\}$ such that $f(m) \neq v_k(m)$ and $f(i) = v_k(i)$ for all $i > m$. We now define $x_i = 0$ for each $i < m$; note that since $0 \in \bar{\mathbb{N}}$, all of these values are in $\bar{\mathbb{N}}$. Since t_i is a linear combination of $x_1, \dots, x_i$, we have that $t_i = 0$ for $i < m$. We define x_m to be any positive number in $\bar{\mathbb{N}}$, resulting in a positive value for t_m, since $x_i = 0$ for $i < m$. Since $f(m) \neq v_k(m)$, we get $t_m f(m) \neq t_m v_k(m)$, and since $f(i) = v_k(i)$ for all $i > m$ we obtain

$$\tau f = \sum_{i=m}^{k} t_i f(i) = t_m f(m) + \sum_{i>m} t_i v_k(i) \neq t_m v_k(m) + \sum_{i>m} t_i v_k(i) = \tau v_k,$$

for all possible values of t_i for $i > m$. So any choice of a positive value for x_m in $\bar{\mathbb{N}}$ ensures that $\tau f \neq \tau v_k$.

For $m < i < k$, we continue choosing positive values for x_i in $\bar{\mathbb{N}}$ that are large enough to ensure that t_i is nonnegative, as explained above. Finally, at the last step, we have chosen $x_1, \dots, x_{k-1}$ and therefore determined the values of $t_1, \dots, t_{k-1}$. Note that furthermore the values of $\tau v_1, \dots, \tau v_{k-1}$ have also been determined: since V_+ is an upper triangular matrix, we have $v_i(j) = 0$ for each $j > i$, so $\tau v_i = t_1 v_i(1) + \cdots + t_i v_i(i) = 0$ for all $i < m$; in particular even though t_k appears in τ, it does not

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

appear in the values of $\tau v_1, \ldots, \tau v_{k-1}$. We now choose $x_k \in \bar{\mathbb{N}}$ so that $t_k > \tau v_i$ for all $i < k$. Since $[f, \mathrm{V}]$ is a basic equation by definition, f is positive in each of its entries and in particular $f(k) \geq 1$, so we obtain $\tau f \geq t_k f(k) \geq t_k > \tau v_i$ for each $i < k$. ⊣

COROLLARY 8.2. *If a simple equation satisfies* $(\star)$, *then it is spineless.*

Thus we obtain the following from Lemma 7.14 and Corollary 8.2:

LEMMA 8.3. *If a simple equation satisfies* $(\star\star)$ *then it is spineless.*

8.3. Prespinality. We will begin with a concrete example of a prespinal equation before illustrating the general case.

EXAMPLE 8.4. Consider the 8-variable simple equation $\varepsilon : stuvwxyz \leq$

$$1 \vee sw^2xz^4 \vee s^2 \vee s^3tx^2yz \vee s^4tz^2 \vee s^5twy^2 \vee s^6z^4 \vee s^7vwx^2y \vee s^8t^2 \vee s^9uvx^2y,$$

where for better readability we use the letters $s, \ldots, z$ for the formal variables $x_1, \ldots, x_8$, in that order; we will use both names for each variable below.

Since ε is an 8-variable equation with 10 distinct joinands, and all spinal equations in 8-variables have no more than 9 distinct joinands, the equation ε is not spinal. However, it is easily verified that ε is prespinal, as witnessed by the 3-variable substitution $\boldsymbol{\sigma}$ defined via: $s \mapsto 1$, $t \mapsto x_1^2$, $z \mapsto x_1$, $y \mapsto x_1x_2$, $v \mapsto x_3$, and $u, w, x \mapsto x_2$. Indeed, we have

$$\boldsymbol{\sigma}\varepsilon : x_1^4x_2^4x_3 \leq 1 \vee x_1^4 \vee x_1^4x_2^3 \vee x_1x_2^4x_3.$$

We name the joinands in the right-hand side of ε in order of appearance from left to right: $d_1 = 1, d_2 = sw^2xz^4, \ldots, d_{10} = s^9uvx^2y$; we do the same for $\boldsymbol{\sigma}\varepsilon$: $v_0 = 1$, $v_1 = x_1^4$, $v_2 = x_1^4x_2^3$, and $v_3 = x_1x_2^4x_3$. Also, we define the sets $\mathrm{D} = \{d_1, \ldots, d_{10}\}$ and $\mathrm{V} = \{v_0, \ldots, v_3\}$. Given this particular ordering of variables and joinands, the set-theoretic equation $\boldsymbol{\sigma}\mathrm{D} = \mathrm{V}$ induces the matrix equation $\boldsymbol{\sigma} D = V$ (note the different font for the sets D, V and the matrices D, V), where

$$\boldsymbol{\sigma} = \begin{array}{c} \begin{array}{cccccccc} s & t & u & v & w & x & y & z \end{array} \\ \begin{bmatrix} 0 & 2 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}, D = \begin{array}{c} \\ s \\ t \\ u \\ v \\ w \\ x \\ y \\ z \end{array} \begin{array}{c} \begin{array}{cccccccccc} d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 & d_8 & d_9 & d_{10} \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 2 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 2 & 0 & 1 & 0 & 1 \\ 0 & 4 & 0 & 1 & 2 & 0 & 4 & 0 & 0 & 0 \end{bmatrix} \end{array}, V = \begin{array}{c} \begin{array}{cccccccccc} v_0 & v_2 & v_0 & v_2 & v_1 & v_2 & v_1 & v_3 & v_1 & v_3 \end{array} \\ \begin{bmatrix} 0 & 4 & 0 & 4 & 4 & 4 & 4 & 1 & 4 & 1 \\ 0 & 3 & 0 & 3 & 0 & 3 & 0 & 4 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}.$$

Note that the (i, j)-entry of D represents the degree of the i-th variable x_i in the joinand d_j, and the (j, i)-entry of $\boldsymbol{\sigma}$ represents the degree of x_j in $\boldsymbol{\sigma}(x_i)$.

Note that, by omitting $v_0 = \mathbf{0}$, the 3×3 matrix $[v_1\ v_2\ v_3]$ is upper triangular with a positive diagonal (as demanded in the definition of $\boldsymbol{\sigma}\varepsilon$ being spinal) and this is the reason for the particular naming of v_0, v_1, v_2, v_3, in that order. In turn, given this order, the substitution partitions the set of joinands D, i.e., the columns of D into $\mathrm{D}_0 = \{d_1, d_3\}$, $\mathrm{D}_1 = \{d_5, d_7, d_9\}$, $\mathrm{D}_2 = \{d_2, d_4, d_6\}$, and $\mathrm{D}_3 = \{d_8, d_{10}\}$, so that $\boldsymbol{\sigma}\mathrm{D}_j = \{v_j\}$, for all j. Guided by this ordering, we further rearrange the columns of V into a new matrix V' and the columns of D into a new matrix D', where the ordering of the columns within each D_i is done randomly. If we represent D'

symbolically as $[\mathrm{D}_0\ \mathrm{D}_1\ \mathrm{D}_2\ \mathrm{D}_3]$, then we also have $\boldsymbol{\sigma} D' = [\boldsymbol{\sigma}\mathrm{D}_0\ \boldsymbol{\sigma}\mathrm{D}_1\ \boldsymbol{\sigma}\mathrm{D}_2\ \boldsymbol{\sigma}\mathrm{D}_3]$ and the equation $\boldsymbol{\sigma} D' = V'$.

We can improve the presentation of this equation even more by putting $\boldsymbol{\sigma}$ and D' in a triangular form, at least in blocks. More specifically, we now rearrange the rows of D' and simultaneously the columns of $\boldsymbol{\sigma}$ (this corresponds to permuting the variables of ε) to obtain new matrices D'' and $\boldsymbol{\sigma}'$, yielding the equation $\boldsymbol{\sigma}' D'' = V'$:

$$
\begin{array}{c}
\begin{array}{cccccccc} s & t & z & w & x & y & u & v \end{array} \\
\left[\begin{array}{c|cc|ccc|cc}
0 & 2 & 1 & 0 & 0 & 1 & 0 & 0 \\ \hline
0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right]
\end{array}
\begin{array}{c}
\begin{array}{cccccccccc} d_1 & d_3 & d_5 & d_7 & d_9 & d_2 & d_4 & d_6 & d_8 & d_{10} \end{array} \\
\left[\begin{array}{cc|ccc|ccc|cc}
0 & 2 & 4 & 6 & 8 & 1 & 3 & 5 & 7 & 9 \\ \hline
0 & 0 & 1 & 0 & 2 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 2 & 4 & 0 & 4 & 1 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 2 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 1 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{array}\right]
\end{array}
=
\begin{array}{c}
\begin{array}{cccccccccc} v_0 & v_0 & v_1 & v_1 & v_1 & v_2 & v_2 & v_2 & v_3 & v_3 \end{array} \\
\left[\begin{array}{cc|ccc|ccc|cc}
0 & 0 & 4 & 4 & 4 & 4 & 4 & 4 & 1 & 1 \\ \hline
0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 4 & 4 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{array}\right]
\end{array}.
$$

Finally, we observe that the rearrangement of the rows results in a partition of the set of rows such that the two partitions (of the set of rows and the set of columns) induce a blocking (given by the solid lines above) that has an upper-triangular shape. We denote by $D^{\flat}$ and $\boldsymbol{\sigma}^{\flat}$ the resulting block matrices, and the equation $\boldsymbol{\sigma}' D'' = V'$ of matrices yields the equation $\boldsymbol{\sigma}^{\flat} D^{\flat} = V^{\flat}$ of block matrices. We call the elements of $D^{\flat}$ *blocks* and they are submatrices of D''; we denote by $(D^{\flat})_{ij}$ the $(i+1, j+1)$-block of $D^{\flat}$, where $0 \leq i, j \leq 3$. We observe that

1. each $(D^{\flat})_{ij}$ is the zero matrix when $i > j$ (all blocks below the diagonal are zero matrices) and
2. each row and each column of $(D^{\flat})_{jj}$ contains a nonzero entry for $j \geq 1$ (in the diagonal blocks no row and no column is fully zero, with the possible exception of the top left block $(D^{\flat})_{00}$).

We call such a partition of the matrix into a block matrix with these two features a *blocking* of the matrix. Each blocking specifies an (ordered) partition of the set of rows and an (ordered) partition of the set of columns of a matrix in the obvious way (with the provision that the first class in this ordered list may be the empty set), but each of these two partitions is special as we will explain. Given a column partition, we define below an associated list of sets of rows. Whenever the original partition comes from a blocking, the resulting list is actually an (ordered) partition (every set in the list, except possibly the first one, is non-empty). The same holds with the roles of rows and columns swapped. We will explain that blockings correspond bijectively to column-partitions that happen to induce ordered row-partitions and to row-partitions that happen to induce ordered column-partitions.

Given an $I \times J$ matrix D, formally viewed as a function from $I \times J$, as usual D_{ij} denotes its entry in the i-th row and j-th column, for $i \in I$ and $j \in J$; usually I and J are taken to be initial segments of the positive integers as in the example above. We denote by $D_{i\_}$ the i-th row and by $D_{\_j}$ the j-th column of D. Given an *ordered partition* $(C_0, C_1, \dots, C_k)$ of J (i.e., C_0 may be empty, but not all of J, and the remaining list forms a partition of J) we define the list $(R_0, R_1, \dots, R_k)$ of subsets of I as follows: for $m \geq 1$, R_m contains those $i \in I$ such that the entry D_{in} is zero for n belonging to parts C_ℓ with $\ell < m$ and there is a non-zero entry D_{in} for some n belonging to the part C_m. In other words, if we group the columns of

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

D according to the ordered partition, then R_m corresponds to those rows that are fully zero on all columns before C_m and are not fully zero on C_m. We define R_0 as containing the remaining i's that are not in $R_1 \cup \cdots \cup R_k$. (Note that we allow our ordered partitions to have an optional initial empty part R_0.) In the example above, the sets R_n are all non-empty, thus resulting into a partition of the set of columns; however, this may not be the case when $(C_0, C_1, \dots, C_k)$ of J is an arbitrary ordered partition of J. Note that whenever $(R_0, R_1, \dots, R_k)$ is a partition, we can define a partition of $I \times J$ into blocks of the form $R_m \times C_n$ with the feature that, for $n \geq 1$, each block $R_n \times C_n$ is such that no column and no row of D in that block is fully zero. Therefore, blockings correspond to column-partitions that happen to induce ordered row-partitions. Often, instead of writing a partition $(C_0, C_1, \dots, C_k)$ of the column index set J, we will be writing the partition $(\mathrm{D}_0, \mathrm{D}_1, \dots, \mathrm{D}_k)$ of the set D of the corresponding columns.

Conversely, given an ordered partition $(R_0, R_1, \dots, R_k)$ of I we define a list $(C_0, C_1, \dots, C_k)$ of subsets of J as follows: for $n \geq 1$, C_n contains those $j \in J$ such that the entry D_{mj} is zero for m belonging to parts R_ℓ with $\ell > n$, and there is a non-zero entry D_{mj} for some m belonging to the part C_n. In other words, if we group the rows of D according to the ordered partition, then R_n corresponds to those rows that finish with zeros on all rows after C_n and are not fully zero on C_n. We define C_0 as containing the remaining i's that are not in $C_1 \cup \cdots \cup C_k$. Again we can see that this yields an ordered partition (i.e., the sets C_n, $n \geq 1$, are non-empty) iff this corresponds to a blocking of the matrix.

Given a blocking $\mathfrak{b}$ of $I \times J$ on a matrix D, we obtain the block matrix $D^{\mathfrak{b}}$ and observe that it has an upper-triangular form. In the following we will consider a blocking given by either its ordered row-partition or its ordered column-partition. In analogy with our notation for entries, rows and columns of a matrix, we define $D^{\mathfrak{b}}_{mn} := \{D_{ij} : i \in R_m, j \in C_n\}$, $D^{\mathfrak{b}}_{m\text{-}} = \{D_{ij} : i \in R_m\}$, and $D^{\mathfrak{b}}_{\text{-}n} = \{D_{ij} : j \in C_n\}$. If D is a set of column vectors, we say $\mathfrak{b}$ is a *blocking of* D if it is a blocking of a matrix D whose set of columns form D, and we define $\mathrm{D}^{\mathfrak{b}}_j$ to be the j-th set of column vectors $D^{\mathfrak{b}}_{\text{-}j}$. Observe that if $\mathfrak{b} = (R_0, \dots, R_k)$ is a row blocking for D, then by how the associated list of columns are defined, we get $\mathrm{D}^{\mathfrak{b}}_j = \{d \in \mathrm{D} : \mathrm{supp}(d) \cap R_j \neq \emptyset\} \setminus (\mathrm{D}^{\mathfrak{b}}_{j+1} \cup \cdots \mathrm{D}^{\mathfrak{b}}_k)$, for each $j \geq 1$, and $\mathrm{D}^{\mathfrak{b}}_0 = \mathrm{D} \setminus (\mathrm{D}^{\mathfrak{b}}_1 \cup \cdots \mathrm{D}^{\mathfrak{b}}_k)$.

The blocking in the Example 8.4 is induced by $\boldsymbol{\sigma}$ in the sense that the corresponding partition on the set $\{1, \dots, 10\}$ of columns of D along the vertical lines in $D^{\mathfrak{b}}$ into the sets $C_0 = \{1, 3\}$, $C_1 = \{5, 7, 9\}$, $C_2 = \{2, 3, 6\}$, and $C_3 = \{8, 10\}$ is given by the stipulation that C_n is exactly the set of the columns that are mapped by $\boldsymbol{\sigma}$ to the same column vector, v_n, of V, for all $n \in \{0, 1, 2, 3\}$. The partition of the set of rows $\{1, \dots, 8\}$ of D (along the horizontal lines of $D^{\mathfrak{b}}$) yields the sets $R_3 = \{3, 4\}$, $R_2 = \{5, 6, 7\}$, $R_1 = \{2, 8\}$, and $R_0 = \{1\}$.

As in the example, every substitution $\boldsymbol{\sigma}$ that maps an $I \times J$ matrix D to a spine $V = [v_0\, v_1\, \cdots\, v_k]$ induces a blocking $\mathfrak{b}$ via a partitioning of the columns: $C_n = \{j \in J : \boldsymbol{\sigma} D_{\text{-}j} = v_n\}$. The blockings induced by substitutions enjoy further properties (in addition to yielding upper-triangular block matrices with diagonal blocks that have no fully zero row or column). Returning to the example we see that $\boldsymbol{\sigma} D^{\mathfrak{b}}_{\text{-}j} = v_j$ for each $0 \leq j \leq 3$. We say that a substitution is a *solution* for a set/matrix of column vectors if it sends all of the vectors of the set/matrix to the same vector. In this

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

terminology, $\boldsymbol{\sigma}$ is a solution for each of the sets $D^{\mathfrak{b}}_{\text{-}0}$, $D^{\mathfrak{b}}_{\text{-}1}$, $D^{\mathfrak{b}}_{\text{-}2}$, and $D^{\mathfrak{b}}_{\text{-}3}$. Note that a substitution is a solution for a set/matrix iff each of its rows is a solution for it.

Also, looking at the induced partition on the rows, the R_m part of each row σ_m of $\boldsymbol{\sigma}$ is not fully zero and all of its elements are non-negative. For $f \in \mathbb{F}$ and $T \subseteq \mathbb{Z}^+$ we say f is *T-positive* if $f_T > \mathbf{0}$, i.e., $f_T \neq \mathbf{0}$ and $f(i) \geq 0$ for each $i \in T$; put differently $f \notin \mathbb{F}_{T^c}$, where T^c is the complement of T. In this terminology, the row σ_m of $\boldsymbol{\sigma}$ is R_m-positive, for each m. Finally, we note that σ_3 is an element of $\mathbb{F}^{\top}_{R_3}$, σ_2 is an element of $\mathbb{F}^{\top}_{R_3 \cup R_2}$, and σ_1 is an element of $\mathbb{F}^{\top}_{R_3 \cup R_2 \cup R_1}$. In general, for every blocking defined by a substitution $\boldsymbol{\sigma}$ with respect to a spine, σ_m is an element of $\mathbb{F}^{\top}_{R^+_m}$, where $R^+_m := R_m \cup \cdots \cup R_k$. For $T \subseteq S$, we say f is (T, S) *-positive* if f is T-positive and $f \in \mathbb{F}_S$; put differently $f \in \mathbb{F}_S$ and $f \notin \mathbb{F}_{T^c}$. Therefore, the row σ_m of $\boldsymbol{\sigma}$ is (R_m, R^+_m)-positive, for each m.

Given a row blocking $\mathfrak{b} = (R_0, \ldots, R_k)$ of a set D, a 1-variable substitution $\sigma \in \mathbb{F}^{\top}$ and $1 \leq i \leq k$, if σ is (R_i, R^+_i)-positive and σ is a solution for each set of columns $\mathrm{D}^{\mathfrak{b}}_{j}$, then we say that σ is a $(\mathfrak{b}, i)$ *-solution* for D. Finally, we say that a k-variable substitution $\boldsymbol{\sigma}$ is a $\mathfrak{b}$ *-solution for* D, where $\mathfrak{b} = (R_0, \ldots, R_k)$, if for all i, the i-th row σ_i of $\boldsymbol{\sigma}$ is a $(\mathfrak{b}, i)$ *-solution* for D.

$$\begin{bmatrix} \mathbf{0} & \boldsymbol{\sigma}^{\mathfrak{b}}_{11} & \cdots & \boldsymbol{\sigma}^{\mathfrak{b}}_{1i} & \cdots & \boldsymbol{\sigma}^{\mathfrak{b}}_{1k} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\sigma}^{\mathfrak{b}}_{ii} & \cdots & \boldsymbol{\sigma}^{\mathfrak{b}}_{ik} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \boldsymbol{\sigma}^{\mathfrak{b}}_{kk} \end{bmatrix} \begin{bmatrix} \mathrm{D}^{\mathfrak{b}}_{00} & \mathrm{D}^{\mathfrak{b}}_{01} & \cdots & \mathrm{D}^{\mathfrak{b}}_{0i} & \cdots & \mathrm{D}^{\mathfrak{b}}_{0k} \\ \mathbf{0} & \mathrm{D}^{\mathfrak{b}}_{11} & \cdots & \mathrm{D}^{\mathfrak{b}}_{1i} & \cdots & \mathrm{D}^{\mathfrak{b}}_{1k} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathrm{D}^{\mathfrak{b}}_{ii} & \cdots & \mathrm{D}^{\mathfrak{b}}_{ik} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathrm{D}^{\mathfrak{b}}_{kk} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & v_1(1) & \cdots & v_i(1) & \cdots & v_k(1) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & v_i(i) & \cdots & v_k(i) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & v_k(k) \end{bmatrix},$$

The following lemma and theorem then follow by the definition of $\mathfrak{b}$-solutions.

LEMMA 8.5. *If a k-variable substitution $\boldsymbol{\sigma}$ is a $\mathfrak{b}$-solution for D, then the matrix $[v_1\ v_2\ \cdots\ v_k]$, where $v_j = \boldsymbol{\sigma} D^{\mathfrak{b}}_{\text{-}j}$, is a $k \times k$ upper-triangular matrix whose diagonal contains positive entries.*

THEOREM 8.6. *An n-variable simple equation* [D] *is prespinal if and only if* D *has a* $\mathfrak{b}$*-solution $\boldsymbol{\sigma}$, for some row blocking* $\mathfrak{b}$*, such that $\boldsymbol{\sigma}\mathbf{1}_n$ and $\boldsymbol{\sigma}\mathrm{D}^{\mathfrak{b}}_{k}$ differ.*

The importance of Theorem 8.6 is that it characterizes the notion of prespinality without a reference to a spine.

Using Theorem 8.6 we now characterize which equations are spineless. For instance, we can verify that Equations (iii)–(v) from Table 1 are spineless. In each equation, the set D has only one blocking $\mathfrak{b}$, given by $(\mathrm{D}^{\mathfrak{b}}_0, \mathrm{D}^{\mathfrak{b}}_1)$ where $\mathrm{D}^{\mathfrak{b}}_1$ contains all the nonzero columns from D. We see that there is no nonzero $(\mathfrak{b}, 1)$-solution for either Equation (iii) or (iv). In the case of Equation (v), the only $(\mathfrak{b}, 1)$-solutions are scalar multiples of $\sigma = [1\ 1\ 1]$, but $\sigma\mathbf{1}_3 \in \sigma\mathrm{D}$. Therefore Equations (iii)–(v) are spineless.

8.4. Spineless equations satisfy $(\star\star)$**.** Theorem 8.6 provides the foundational link between an equation being spineless and satisfying $(\star\star)$, namely by the (non-) existence of certain 1-variable substitutions viewed as solutions to particular linear systems.

We prove that every spineless equation satisfies $(\star\star K)$, for sufficiently large K, contrapositively. So, we assume that a simple equation [D] satisfies the antecedent

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

of $(\star\star K)$ and in particular there exists a (nontrivial) 1-variable substitution σ for which $\sigma\mathrm{D}$ is contained in some shift of $K^{\mathbb{N}}$. The following lemma ensures that, if K is chosen large enough, σ *induces* a blocking $\mathfrak{b}$ on D in the sense that the naturally ordered column-partition $(\mathrm{D}_0, \dots, \mathrm{D}_k)$ of D induces $\mathfrak{b}$, i.e., $\mathrm{D}_j = \mathrm{D}_j^{\mathfrak{b}}$ for each j. Here we conventionally order the sets so that $\sigma\mathrm{D}_j < \sigma\mathrm{D}_{j+1}$ for $j < k$, and we always take D_0 to be the (possibly empty) set of all $d \in \mathrm{D}$ such that $\sigma d = 0$. Note that if a σ induces a blocking then it must be that $k \geq 1$ and so σ is not the zero vector.

For a finite set $\mathrm{D} \subseteq \mathbb{F}$, we define $\Delta\mathrm{D} = \sum_{i=1}^{n} \max\{|d(i) - d'(i)| : d, d' \in \mathrm{D}\}$.

LEMMA 8.7. *Assume that for a finite* $\mathrm{D} \subseteq \mathbb{F}$ *and for some* 1*-variable substitution* σ *that is nontrivial on* D, $\sigma\mathrm{D}$ *is contained in some shift of* $K^{\mathbb{N}}$, *where* $K > \Delta\mathrm{D} + 1$. *Then* σ *induces a blocking on* D.

PROOF. Suppose $\mathrm{D} \subseteq \mathbb{F}_n$ and let $(\mathrm{D}_0, \dots, \mathrm{D}_k)$ be the column partition of D such that $0 = \sigma\mathrm{D}_0 < \cdots < \sigma\mathrm{D}_k$. We will show that the associated list $(R_0, R_1, \dots, R_k)$ of sets of rows is an ordered row-partition, i.e., each of the $R_1, \dots, R_k$ is not empty. Recall that R_j is the set of all indices $i \in \{1, \dots, n\}$ such that $D_{ij} \neq \mathbf{0}$ but $D_{il} = \mathbf{0}$ for each $l < j$, where $D := [\mathrm{D}_0 \cdots \mathrm{D}_k]$ (the order of the columns within each D_i plays no role). We first prove that R_k induces D_k, i.e., each $d \in \mathrm{D}_k$ must have some nonzero entry/row that is zero in every $d' \in \mathrm{D} \setminus \mathrm{D}_k$. If not, then there exists some $d \in \mathrm{D}_k$ such that for each $i \in \mathrm{supp}(d)$ there exists $j < k$ and vector $d' \in \mathrm{D}_j$ with $i \in \mathrm{supp}(d')$. By assumption, σd and $\sigma d'$ are in the same shift of $K^{\mathbb{N}}$, say $K^{\mathbb{N}} - C$ for some $C \in \mathbb{N}$, and because of our ordering convention, $\sigma d' < \sigma d$. So, $\sigma d = K^{a+1} - C$ and $\sigma d' \leq K^a - C$ for some $a \geq 0$. Since $d'(i) \geq 1$, we have $\sigma(i) \leq \sigma(i)d'(i) \leq \sigma d' \leq K^a$, so

$$K^a(K-1) = K^{a+1} - K^a \leq \sigma d - \sigma d' \leq \sigma|d - d'| \leq K^a \Delta\{d, d'\} \leq K^a \Delta\mathrm{D},$$

where the entries of $|d - d'|$ are the absolute values of the corresponding entries of $d - d'$. Therefore, $K \leq \Delta\mathrm{D} + 1$, which contradicts the assumption on the size of K. So, R_k is nonempty and we obtain $\mathrm{D}_k = \{d \in \mathrm{D} : \mathrm{supp}(d) \cap R_k \neq \emptyset\}$.

Continuing in this way for $1 \leq j < k$, set $\mathrm{D}' = \mathrm{D}_0 \cup \cdots \cup \mathrm{D}_j$. Since $\mathrm{D}' \subseteq \mathrm{D}$ implies $\Delta\mathrm{D}' \leq \Delta\mathrm{D}$, the same argument shows each $d \in \mathrm{D}_j$ contains a nonzero entry that is zero for each $d' \in \mathrm{D}' \setminus \mathrm{D}_j$. Hence R_j induces D_j, i.e.,

$$\mathrm{D}_j = \{d \in \mathrm{D} : \mathrm{supp}(d) \cap R_j \neq \emptyset\} \setminus (\mathrm{D}_{j+1} \cup \cdots \cup \mathrm{D}_k).$$

Given $R_0 = \mathrm{supp}(\mathrm{D}) \setminus R_1^+$ by definition, we conclude that $\mathfrak{b} = (R_0, \dots, R_k)$ is a blocking on D such that $\mathrm{D}_j^{\mathfrak{b}} = \mathrm{D}_j$ for each $j \leq k$, i.e., σ induces the blocking $\mathfrak{b}$ on D. ⊣

Suppose σ induces a row blocking $\mathfrak{b} = (R_0, R_1, \dots, R_k)$ of D. Since $\sigma\mathrm{D}_1 > 0$ by definition, it must be that σ is R_1-positive, and since $\sigma\mathrm{D}_0 = 0$ by definition, it must be that the support σ is contained in R_1^+. That is, σ must be (R_1, R_1^+)-positive. Since it is also a solution for each $\mathrm{D}_j^{\mathfrak{b}}$ by definition, this proves the following lemma.

LEMMA 8.8. *If a* 1*-variable substitution* σ *induces a blocking* $\mathfrak{b}$ *on* D, *then it is a* $(\mathfrak{b}, 1)$*-solution for* D. *In particular, if* $\boldsymbol{\sigma}$ *is any* $\mathfrak{b}$*-solution for* D, *the substitution obtained by replacing the first row of* $\boldsymbol{\sigma}$ *by* σ *is also a* $\mathfrak{b}$*-solution for* D.

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

We now demonstrate the converse of Corollary 8.3 by proving the following stronger statement. The proof relies on Lemma 8.13, which we prove in the next section.

LEMMA 8.9. *A spineless equation satisfies* ($\star\star K$) *for all sufficiently large* K.

PROOF. Toward establishing the contrapositive, we assume that the n-variable simple equation [D] fails ($\star\star K$) for infinitely many $K \in \mathbb{N}$. Then there are infinitely many pairs of 1-variable substitutions σ, σ' witnessing such failures that furthermore induce pairs of blockings for D by Lemma 8.7. Since there are only finitely many blockings, and thus finitely many pairs of them, there must exist blockings $\mathfrak{b} = (R_0, \dots, R_k)$ and $\mathfrak{c} = (R'_0, \dots, R'_l)$ that are witnessed infinitely often as a pair. By Lemma 8.13, [D] has a $\mathfrak{b}$-solution $\boldsymbol{\sigma}$ and a $\mathfrak{c}$-solution $\boldsymbol{\sigma}'$. By Lemma 8.8, we can replace the first rows of $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}'$ with 1-variable substitutions σ_1 and σ'_1 (from the above-mentioned infinitely many) witnessing the failure of ($\star\star K$) and inducing $\mathfrak{b}$ and $\mathfrak{c}$, respectively.

If either $\boldsymbol{\sigma}\mathbf{1}_n \neq \boldsymbol{\sigma}\mathrm{D}_k^{\mathfrak{b}}$ or $\boldsymbol{\sigma}'\mathbf{1}_n \neq \boldsymbol{\sigma}'\mathrm{D}_l^{\mathfrak{c}}$, then [D] is prespinal by Theorem 8.6, and we are done. If not, we have $\boldsymbol{\sigma}\mathbf{1}_n = \boldsymbol{\sigma}\mathrm{D}_k^{\mathfrak{b}}$ and likewise $\boldsymbol{\sigma}'\mathbf{1}_n = \boldsymbol{\sigma}'\mathrm{D}_l^{\mathfrak{c}}$. In particular, $\sigma_1\mathbf{1}_n = \sigma_1 d$ for every $d \in \mathrm{D}_k^{\mathfrak{b}}$ and $\sigma'_1\mathbf{1}_n = \sigma'_1 d$ for every $d \in \mathrm{D}_l^{\mathfrak{c}}$, and hence for every $d \in \mathrm{D}_k^{\mathfrak{b}} \cap \mathrm{D}_l^{\mathfrak{c}}$. If $\mathrm{D}_k^{\mathfrak{b}}$ and $\mathrm{D}_l^{\mathfrak{c}}$ were not disjoint, there would be a $d \in \mathrm{D}$ such that $\sigma_1\mathbf{1}_n = \sigma_1 d$ and $\sigma'_1\mathbf{1}_n = \sigma'_1 d$, which would imply that σ_1, σ'_1 satisfy ($\star\star K$), contradicting their choice above. Hence $\mathrm{D}_k^{\mathfrak{b}}$ and $\mathrm{D}_l^{\mathfrak{c}}$ are disjoint and by the definition of blockings, R_k and R'_l are also disjoint.

As a result the row partition $\mathfrak{a} := (B_0, B_1)$, where $B_1 = R_k \cup R'_l$ and $B_0 = \{1, \dots, n\} \setminus B_1$, is actually a row blocking on D which furthermore induces the partition of columns $\mathrm{D}_1^{\mathfrak{a}} = \mathrm{D}_k^{\mathfrak{b}} \cup \mathrm{D}_l^{\mathfrak{c}}$ and $\mathrm{D}_0^{\mathfrak{a}} = \mathrm{D} \setminus \mathrm{D}_1^{\mathfrak{a}}$. We will construct a 1-variable substitution α that will serve as an $\mathfrak{a}$-solution for D witnessing the prespinality of [D].

Let σ_k and σ'_l be the bottom rows of $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}'$, respectively. Since σ_k is a (R_k, R_k^+)-positive solution for $\mathrm{D}_k^{\mathfrak{b}}$, the value $t = \sigma_k \mathrm{D}_k^{\mathfrak{b}}$ is positive, and since R_k and R'_l are disjoint, $\sigma_k \mathrm{D}_l^{\mathfrak{c}}$ is zero; in detail R'_l contains columns that appear in earlier blocks than R_k. Similarly, $t' = \sigma'_l \mathrm{D}_l^{\mathfrak{c}}$ is positive and $\sigma'_l \mathrm{D}_k^{\mathfrak{b}}$ is zero. Note that $\sigma_k \mathbf{1}_n = t$ and $\sigma'_l \mathbf{1}_n = t'$ follow by the fact that $\boldsymbol{\sigma}\mathbf{1}_n = \boldsymbol{\sigma}\mathrm{D}_k^{\mathfrak{b}}$ and $\boldsymbol{\sigma}'\mathbf{1}_n = \boldsymbol{\sigma}'\mathrm{D}_l^{\mathfrak{c}}$.

We now define the 1-variable substitution $\alpha = t'\sigma_k + t\sigma'_l$. Since R_k and R'_l are disjoint, it follows that $\alpha\mathrm{D}_k^{\mathfrak{b}} = t't + t0 = tt'$ and $\alpha\mathrm{D}_l^{\mathfrak{c}} = t'0 + tt' = tt'$, so $\alpha\mathrm{D}_1^{\mathfrak{a}} = tt'$; also $\alpha\mathbf{1}_n = t't + tt' = 2tt'$. In the case when $\mathrm{D}_0^{\mathfrak{a}}$ is nonempty, we have $\alpha\mathrm{D}_0^{\mathfrak{a}} = 0$, because $\sigma_k \mathrm{D}_j^{\mathfrak{b}} = 0$, for $j < k$, and $\sigma'_l \mathrm{D}_i^{\mathfrak{c}} = 0$, for $i < l$. Since $t, t' > 0$, we have $2tt' > tt' > 0$, and therefore it follows that $\alpha\mathbf{1}_n \notin \alpha\mathrm{D}$ and so [D] is prespinal by Theorem 8.6. ⊣

Lemmas 8.3 and 8.9 establish the equivalences between a simple equation being spineless and satisfying ($\star\star$), and as well as satisfying ($\star$).

THEOREM 8.10. *A simple equation is spineless iff it satisfies* ($\star\star K$) *for every sufficiently large* K.

Therefore, to establish Theorem 8.10 we must prove Lemma 8.13.

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

8.5. Solutions in $\mathbb{R}^n$. The goal of this section is to prove Lemma 8.13. To address this, we recall a theorem of alternatives for positive solutions to linear systems accompanied by conventional terminology (see [15]).

Let $v \in \mathbb{R}^n$ (viewed as a row vector) and $M \subseteq \mathbb{R}^n$. We say that v is *orthogonal* to M if $vM = 0$. We say that v is *strictly positive* if $v \neq \mathbf{0}$ and $v(i) \geq 0$ for each $i \in \{1, \dots, n\}$. The set X^n_+ denotes the set of all strictly positive vectors in X^n, called the *strictly positive orthant* in X^n, where $X \in \{\mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$. The following (folklore) theorem is equivalent to Farkas's Lemma (for instance, see [14, Theorem 27]).

THEOREM 8.11. *Let $M \subseteq \mathbb{R}^n$ be nonempty set of vectors and $i \in \{1, \dots, n\}$ a fixed index. Then exactly one of the following holds*:

1. *there exists a strictly positive vector v orthogonal to M where $v(i) > 0$, or*
2. *there exists a strictly positive vector $w \in \operatorname{span}(M)$ where $w(i) > 0$.*

Note that $\operatorname{span}(M)_S = \operatorname{span}(M_S)$ for any $M \subseteq \mathbb{R}^n$ and $S \subseteq \{1, \dots, n\}$; here the subscript S denotes restriction to S.

COROLLARY 8.12. *Let $M \subseteq \mathbb{R}^n$ and $T \subseteq S \subseteq \{1, \dots, n\}$ be non-empty. If there is no T-positive vector in $\mathbb{R}^{|S|}_+$ orthogonal to M_S then there exists $L \in \mathbb{N}$ such that, for any $v \in \mathbb{R}^n_+$ orthogonal to M, $v(i) \leq L \cdot \max\{v(j) : j \in \{1, \dots, n\} \setminus S\}$ for each $i \in T$.*

PROOF. By Theorem 8.11, for each $i \in T$ there exists a strictly positive (in $\mathbb{R}^{|S|}$) vector $w_i \in \operatorname{span}(M_S)$ where $w_i(i) > 0$. Then $\bar{w} := \sum_{i \in T} w_i \in \operatorname{span}(M_S)$ is strictly positive with $T \subseteq \operatorname{supp}(\bar{w})$. Let $w \in \operatorname{span}(M)$ be such that $w_S = \bar{w}$. Note that since T is nonempty and $T \subseteq \operatorname{supp}(w_S)$, it follows that $t := \min\{w(i) : i \in T\}$ is positive. Set $S^c := \{1, \dots, n\} \setminus S$ and $m := \sum_{j \in S^c} |w(j)|$, where we take the empty sum to be zero. Define L to be the smallest positive integer greater than m/t.

Now, suppose $v \in \mathbb{R}^n_+$ is orthogonal to M. Then $vw^\top = 0$ and hence

$$\sum_{j \in S} w(j)v(j) = \sum_{j \in S^c} -w(j)v(j).$$

Let N denote this common value. Considering the right-hand side of the equation, we obtain $N \leq m \cdot \max\{v(j) : j \in S^c\}$. Considering the left-hand side of the equation, $tv(i) \leq w(i)v(i) \leq N$, for all $i \in T$, since $T \subseteq S$ and $w(i) \geq t > 0$. Therefore, for all $i \in T$, we deduce $v(i) \leq N/t \leq L \cdot \max\{v(j) : j \in S^c\}$. $\dashv$

LEMMA 8.13. *Let D be a finite subset of $\mathbb{F}_n$ and $\mathfrak{b}$ be a blocking on D. If there are infinitely many $K \in \mathbb{N}$ for which there exists a 1-variable substitution σ that induces $\mathfrak{b}$ and $\sigma\mathrm{D}$ is contained in some shift of $K^{\mathbb{N}}$, then D has a $\mathfrak{b}$-solution.*

PROOF. Working contrapositively, we assume that D has no $\mathfrak{b}$-solution. We will show that if $K > 1$ is such that there exists a 1-variable substitution $\sigma \in \mathbb{F}_n^\top$ that induces a blocking $\mathfrak{b}$ on D and that $\sigma\mathrm{D}$ is contained in some shift of $K^{\mathbb{N}}$, then K can be no larger than a certain multiple of $\Delta\mathrm{D}$. Let $\mathfrak{b} = (R_0, \dots, R_k)$, where $k \geq 1$.

For any nonempty $A \subseteq \mathbb{F}_n$, fix $\bar{a} \in A$ and define $\bar{A} := \{a - \bar{a} : a \in A\}$; note that the entries of the column vectors are in $\mathbb{Z}$. For a set of rows $S \subseteq \{1, \dots, n\}$, σ is a solution for A_S iff σA_S is a singleton iff $\sigma A_S = \sigma\bar{a}$ iff $\sigma\bar{A}_S = \{0\}$ iff σ is orthogonal to $\bar{A}_S$ in $\mathbb{R}^n$ (regardless of the choice $\bar{a} \in A$). Hence, if $T \subseteq S$, then there exists a

T-positive solution for A in $\mathbb{F}_S^{\top}$ iff there exists a T-positive solution for A_S iff there exists a T-positive vector of $\mathbb{R}_+^{|S|}$ orthogonal to $\bar{A}_S$.[15]

Now, by definition of being a blocking, for each $i \geq k$ the set $\mathrm{D}_i^{\mathfrak{b}}$ is nonempty, so we can define $\bar{\mathrm{D}}_i^{\mathfrak{b}} = \{d - \bar{d}_i : d \in \mathrm{D}_i^{\mathfrak{b}}\}$ for some fixed $\bar{d}_i \in \mathrm{D}_i^{\mathfrak{b}}$. Since, if $\mathbf{0} \in \mathrm{D}$ then $\mathbf{0} \in \mathrm{D}_0^{\mathfrak{b}}$ by definition, we may define the (possibly empty) set $\bar{\mathrm{D}}_0^{\mathfrak{b}} := \mathrm{D}_0^{\mathfrak{b}}$. We note that if σ is a $(\mathfrak{b}, i)$-solution for D then $\sigma\bar{\mathrm{D}}^{\mathfrak{b}} = 0$, where $\bar{\mathrm{D}}^{\mathfrak{b}} := \bar{\mathrm{D}}_0^{\mathfrak{b}} \cup \cdots \cup \bar{\mathrm{D}}_k^{\mathfrak{b}}$.

Since D has no $\mathfrak{b}$-solution, there must be some $1 \leq i \leq n$ for which there is no $(\mathfrak{b}, i)$-solution. However, since σ induces $\mathfrak{b}$, Lemma 8.8 implies that σ is a $(\mathfrak{b}, 1)$-solution, and so $i > 1$. Therefore, for some $i > 1$, there is no R_i-positive $v \in \mathbb{R}_+^{|S|}$ orthogonal to M_S, where $M = \bar{\mathrm{D}}^{\mathfrak{b}}$ and $S = R_i^+$. Since σ induces a blocking, σ is strictly positive, and since σ is a $(\mathfrak{b}, 1)$-solution for D, σ is orthogonal to $\bar{\mathrm{D}}^{\mathfrak{b}}$; since further $R_i \subseteq R_i^+$, by Corollary 8.12 we have that there exists $L \in \mathbb{N}$ such that $\sigma(t) \leq L \cdot \max\{\sigma(x) : x \in X\}$ for all $t \in R_i$, where $X := \{1, \ldots, n\} \setminus R_i^+ = R_0 \cup \ldots \cup R_{i-1}$.

Since σ induces $\mathfrak{b}$ and since $i > 1$, we have $0 < \sigma\mathrm{D}_{i-1}^{\mathfrak{b}} < \sigma\mathrm{D}_i^{\mathfrak{b}}$. As $\sigma\mathrm{D}$ is contained in a shift of $K^{\mathbb{N}}$, say $K^{\mathbb{N}} - C$ for some $C \in \mathbb{N}$, there must be $a \geq 0$ and $b > 0$ such that $\sigma\mathrm{D}_{i-1}^{\mathfrak{b}} = K^a - C$ and $\sigma\mathrm{D}_i^{\mathfrak{b}} = K^{a+b} - C$. Observe that $\sigma(x) \leq K^a$ for all $x \in X$ since $\sigma\mathrm{D}_j^{\mathfrak{b}} \leq \sigma\mathrm{D}_{i-1}^{\mathfrak{b}} \leq K^a$ for each $j \leq i - 1$ by definition of σ inducing $\mathfrak{b}$. Hence $\sigma(t) \leq LK^a$ for all $t \in X \cup R_i$.

For $d \in \mathrm{D}_i^{\mathfrak{b}}$ and $d' \in \mathrm{D}_{i-1}^{\mathfrak{b}}$, we have $\mathrm{supp}(\{d, d'\}) \subseteq X \cup R_i$, so

$$K^a(K - 1) \leq K^a(K^b - 1) = \sigma d - \sigma d' \leq \sigma|d - d'| \leq LK^a\Delta\{d, d'\} \leq LK^a\Delta\mathrm{D}.$$

It follows that $K \leq L\Delta\mathrm{D} + 1$. ⊣

The lemma above completes the proof of Lemma 8.9 and hence Theorem 8.10.

Now, if Γ is a finite set of spineless simple equations then each equation in Γ must be non-mingly by Lemma 6.4, and furthermore there must exist a smallest K for which each equation in Γ satisfies $(\star\star K)$ as a consequence of Theorem 8.10. Therefore, by Corollary 7.16, we obtain:

COROLLARY 8.14. *For any finite set of spineless simple equations* Γ, *every subvariety of* RL *containing* $\mathsf{CRL} + \Gamma$ *has undecidable word problem.*

This completes the proof of Theorem 5.5, and therefore also of Theorem 5.9.

§9. Concluding remarks. First, we note that the quasiequations used to establish Theorem 4.5 are in the signature $\{\vee, \cdot, 1\}$, so all complexity lower-bound/undecidability results hold even when restricting the word problem to the $\{\vee, \cdot, 1\}$-fragments of such varieties. On the other hand, the equations used to establish Theorem 5.9 make use of the full signature.

COROLLARY 9.1. *Let* ε *be a spineless equation that is simple over* RL *and* $\mathcal{V}$ *a variety of residuated lattices containing* CRL_ε *as a subvariety. Then the word problem*

[15]For the reverse direction, if $v \in \mathbb{R}_+^{|S|}$ is orthogonal to $\bar{A}_S$, then since $\bar{A}_S$ has integer entries, by Gaussian Elimination we may assume that $v \in \mathbb{Q}_+^{|S|}$, and so $t \cdot v \in \mathbb{Z}_+^{|S|}$ for some $t \in \mathbb{N}$ and $t \cdot v$ is orthogonal to $\bar{A}_S$.

(and hence quasiequational theory) for the $\{\vee, \cdot, 1\}$-fragment of $\mathcal{V}$ is undecidable. Furthermore, if ε is expansive then the equational theory for $\mathcal{V}$ is undecidable.

Given a simple equation ε, by (ε) we denote its corresponding sequent-style inference rule, e.g., if $\varepsilon : xy \leq x^2yx \vee x \vee 1$, then

$$\frac{\Gamma, \Delta_1, \Delta_1, \Delta_2, \Delta_1\Sigma \vdash \Pi \quad \Gamma, \Delta_1, \Sigma \vdash \Pi \quad \Gamma, \Sigma \vdash \Pi}{\Gamma, \Delta_1, \Delta_2, \Sigma \vdash \Pi} \ (\varepsilon).$$

COROLLARY 9.2. *Let ε be a spineless simple equation and* **L** *any logic contained in the interval from* $\mathbf{FL_e} + (\varepsilon)$ *to* **FL**. *Then deducibility in the $\{\vee, \cdot, 1\}$-fragment of* **L** *is undecidable. Furthermore, if ε is expansive then provability in the (0-free fragment of)* **L** *is undecidable.*

9.1. Commutative varieties and single variable extensions. In the following table we display decidability results for subvarieties of CRL axiomatized by 1-variable equations using Lemma 5.4. The numbers $n, p, q, \ldots$ are distinct and positive, $m \geq 0$, and furthermore are all given so that ε is not trivial. By $(1 \vee)$ we mean 1 may or may not be included in the expression.

ε	Eq. Th. of CRL_ε	Quasi-Eq Th. of CRL_ε
$x^n \leq x^m$	FMP [17]	FEP [17]
$x^n \leq x^m \vee 1$	?	?
$x^n \leq (1 \vee)\, x^p \vee x^q \vee \cdots$	?	Und. (Thm 5.5)
$x^n \leq x^{n+p} \vee x^{n+q} \vee \cdots$	Und. (Thm 5.9)	Und. (Thm 5.5)

We note that subvarieties axiomatized by equations of the form $x^n \leq x \vee 1$ have the finite model property. In fact, any subvariety of RL axiomatized by a simple equation in which each term on the right-hand side is linear (i.e., any variable occurs at most once in any joinand) has the finite model property (see Theorem 3.15 in [5]). Similarly, while the subvariety of CRL axiomatized by the simple equation $\varepsilon : xyz \leq xy \vee xz \vee yz \vee x \vee y \vee z$ has an undecidable word problem (it is easily verified that ε is spineless), $\mathsf{CRL} + \varepsilon$ has the FMP.

Subvarieties of CRL axiomatized by equations of the form $x^n \leq x^{m+1} \vee 1$, the simplest of which is $\mathsf{d} : x \leq x^2 \vee 1$, have no known decidability results for their (quasi-)equational theories. Focusing on the equation d, we make the following observations:

- $\mathsf{CRL_d}$ does not have the finite embeddability property. In fact, extensions of CRL by equations of the form $x^n y^m \leq x^{2n} \vee y^{2m}$ do not have the FEP for any choice n, m. This follows from the fact that such equations hold in chains and the FEP fails in such extensions of CRL (see [7]).
- The quasiequational theory of $\mathsf{CRL_d}$ does not have a primitive recursive decision procedure. This can be shown using Theorem 4.5 and the machine constructed in [16] (which shows that provability in $\mathbf{FL_{ec}}$, while decidable, is not primitive recursive). In fact, the same construction can be used to show that there is no primitive recursive decision procedure for the quasiequational theory of the subvariety of CRL axiomatized by $x^m \leq x^{m+n}$ $(\vee\ 1)$ with $m, n > 1$.

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

Furthermore, by Corollary 5.8 the same holds for the equational theory of the subvariety of CRL axiomatized by $x^m \leq x^{m+n}$, as this equation is expansive. A more general treatment will be given in a forthcoming paper.

9.2. Non-commutative varieties. As mentioned above, although the main reason for our study has been to establish undecidability for commutative varieties, we also get results about non-commutative ones. Here we compare that portion of our results with existing ones. In [8], it is shown that any variety of residuated lattices containing (as a subvariety) $\mathcal{H} = \mathsf{RL} + (x^3 \leq x^2) + (x \leq x^2)$ has an undecidable word problem witnessed in its $\{\leq, \cdot, 1\}$-fragment; of course no subvariety of CRL contains $\mathcal{H}$, so this result has no implication about commutative varieties. In fact, the algebra $\mathbf{W}^+ \in \mathcal{H}$ constructed in [8] satisfies every equation for which the deletion of any collection of its variables results in either a trivial equation or one with the right-hand side containing a *square* subterm (i.e., a joinand of the from uv^2w for $u, v, w \in X^*$ with $v \neq 1$.) As a result, even though Theorem 5.5 covers a lot of commutative varieties, it does not offer any new results for non-commutative varieties axiomatized by 1-variable equations. In fact, the results in [8] entail undecidability of the word problem for many non-commutative extensions of RL by prespinal equations; e.g., $\mathsf{RL_d}$ has an undecidable word problem since $\mathbf{W}^+ \models x \leq x^2 \vee 1$.

However, there are infinitely many equations in two or more variables for which Theorem 5.5 is applicable while [8] is not. For example, any equation that is rendered trivial via commutativity (e.g., $x^2y \leq xyx$) is spineless and hence subvarieties of RL by such equations have an undecidable word problem by Theorem 5.5.

More interesting 3-variable basic equations can be obtained by using *square-free* joinands.[16] Let $X = \{x, y, z\}$ and let $h : X^* \to X^*$ be the homomorphism extending the assignment $h(1) = 1$, $h(x) = xyz$, $h(y) = xz$, and $h(z) = y$. One can produce square-free words of arbitrary length by considering $h^0(x) = x$ and $h^{k+1}(x) = h(h^k(x))$ (see [12]). For any nontrivial 1-variable equation $\varepsilon : x^n \leq (1 \vee)\, x^p \vee x^q \vee \cdots$, we denote by $h(\varepsilon)$ the basic equation:

$$h^n(x) \leq (1 \vee)\, h^p(x) \vee h^q(x) \vee \cdots .$$

Since ε is nontrivial, $h(\varepsilon)$ is nontrivial and furthermore has square-free joinands. Consequently, $\mathcal{H} \not\models h(\varepsilon)$ and so [8] is not applicable to equations of this form. However, if ε is a spineless 1-variable basic equation then it can easily be shown that $h(\varepsilon)_{\mathcal{C}}$ is also spineless and therefore $\mathsf{RL} + h(\varepsilon)$ has an undecidable word problem by Theorem 5.5. For example, consider the equation $\varepsilon : x \leq x^2 \vee x^3$, then

$$h(\varepsilon) : xyz \leq xyzxzy \vee xyzxzyxyzyxz \quad \text{and} \quad h(\varepsilon)_{\mathcal{C}} : xyz \leq x^2y^2z^2 \vee x^4y^4z^4.$$

It is easily checked using Theorem 8.6 that $h(\varepsilon)_{\mathcal{C}}$ is spineless and hence $h(\varepsilon)$ is spineless in view of Definition 5.3. Therefore $\mathsf{RL} + h(\varepsilon)$ has an undecidable word problem by Theorem 5.5, but $\mathcal{H} \not\models h(\varepsilon)$.

While our undecidability results for the word problem in these varieties takes place in the $\{\vee, \cdot, 1\}$-fragment, we can strengthen such results to the $\{\leq, \cdot, 1\}$-fragment for many non-commutative extensions of residuated lattices (even by some prespinal

[16] A word $w \in X^*$ is square-free if $w \neq ux^2v$ for any words $u, v, x \in X^*$ and $x \neq 1$.

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press

equations) using a different encoding not relying on the $\vee$ operation. Such ideas will be explored in a forthcoming paper.

REFERENCES

[1] W. Blok and D. Pigozzi, *Algebraizable logics*. ***Memoirs of the American Mathematical Society***, vol. 77 (1989), no. 396, pp. 1–78.

[2] R. Cardona and N. Galatos, *The FEP for some varieties of fully distributive knotted residuated lattices*. ***Algebra Universalis***, vol. 78 (2017), no. 3, pp. 363–376.

[3] K. Chvalovský and R. Horčík, *Full lambek calculus with contraction is undecidable*, this Journal, vol. 81 (2016), no. 2, pp. 524–540.

[4] A. Ciabattoni, N. Galatos, and K. Terui, *Algebraic proof theory for substructural logics: Cut-elimination and completions*. ***Annals of Pure and Applied Logic***, vol. 163 (2012), no. 3, pp. 266–290.

[5] N. Galatos and P. Jipsen, *Residuated frames with applications to decidability*. ***Transactions of the American Mathematical Society***, vol. 365 (2013), no. 3, pp. 1219–1249.

[6] N. Galatos, P. Jipsen, T. Kowalski, and H. Ono, ***Residuated Lattices: An Algebraic Glimpse at Substructural Logics***, Studies in Logic and the Foundations of Mathematics, vol. 151, Elsevier, Amsterdam, 2007.

[7] J. Hart, L. Rafter, and C. Tsinakis, *The structure of commutative residuated lattices*. ***International Journal of Algebra and Computation***, vol. 12 (2002), pp. 509–524.

[8] R. Horčík, *Word problem for knotted residuated lattices*. ***Journal of Pure Applied Algebra***, vol. 219 (2015), pp. 1548–1563.

[9] P. Jipsen and C. Tsinakis, *A survey of residuated lattices*, ***Ordered Algebraic Structures*** (J. Martinez, editor), Developmental Mathematics, vol. 7, Kluwer Academic Publishers, Dordrecht, 2002, pp. 19–56.

[10] J. Lambek, *How to program an infinite abacus*. ***Canadian Mathematical Bulletin***, vol. 4 (1961), pp. 265–302.

[11] P. Lincoln, J. Mitchell, A. Scedrov, and N. Shankar, *Decision problems for propositional linear logic*. ***Annals of Pure and Applied Logic***, vol. 56 (1992), pp. 239–311.

[12] M. Lothaire, ***Algebraic Combinatorics on Words***, Cambridge University Press, Cambridge, 2002.

[13] M. Minksy, *Recursive unsolvability of Post's problem of 'tag' and other topics in the theory of Turing machines*. ***The Annals of Mathematics***, vol. 74 (1961), pp. 437–455.

[14] C. Perng, *On a class of theorems equivalent to Farkas's lemma*. ***Applied Mathematical Sciences***, vol. 11 (2017), no. 44, pp. 2175–2184.

[15] S. Roman, ***Positive Solutions to Linear Systems: Convexity and Separation***, Springer, New York, 2005.

[16] A. Urquhart, *The complexity of decision procedures in relevance logic II*, this Journal, vol. 64 (1999), no. 4, pp. 1774–1802.

[17] C. J. van Alten, *The finite model property for knotted extensions of propositional linear logic*, this Journal, vol. 70 (2005), no. 1, pp. 84–98.

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF DENVER
DENVER, CO 80210, USA
E-mail: Nikolaos.Galatos@du.edu

DIPARTIMENTO DI PEDAGOGIA,
PSICOLOGIA, & FILOSOFIA
UNIVERSITÀ DI CAGLIARI
09123 CAGLIARI, ITALY
E-mail: Gavin.StJohn@gmail.com

https://doi.org/10.1017/jsl.2021.46 Published online by Cambridge University Press