BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

SOME EXTENSIONS OF KLEENE ALGEBRA AND THEIR APPLICATIONS

by

CHENG ZHANG

B.A., Wheaton College, 2018

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2024

MS theses only: Prior to having this page signed by the readers, please have it reviewed by the Mugar Library staff. This does not apply to PhD dissertations since this page is completed through DocuSign. **Remove this comment in the final document** by commenting out `\approvalpagewithcomment` and uncommenting `\approvalpage` statements in the *prelim.tex* file.

Approved by

First Reader

First M. Last, PhD
Professor of Electrical and Computer Engineering

Second Reader

First M. Last
Associate Professor of …

Third Reader

First M. Last
Assistant Professor of …

Facilis descensus Averni;
Noctes atque dies patet atri janua Ditis;
Sed revocare gradum, superasque evadere ad auras,
Hoc opus, hic labor est. Virgil (from Don's thesis!)

Acknowledgments

Here go all your acknowledgments. You know, your advisor, funding agency, lab mates, etc., and of course your family.

As for me, I would like to thank Jonathan Polimeni for cleaning up old LaTeX style files and templates so that Engineering students would not have to suffer typesetting dissertations in MS Word. Also, I would like to thank IDS/ISS group (ECE) and CV/CNS lab graduates for their contributions and tweaks to this scheme over the years (after many frustrations when preparing their final document for BU library). In particular, I would like to thank Limor Martin who has helped with the transition to PDF-only dissertation format (no more printing hardcopies – hooray !!!)

The stylistic and aesthetic conventions implemented in this LaTeX thesis/dissertation format would not have been possible without the help from Brendan McDermot of Mugar library and Martha Wellman of CAS.

Finally, credit is due to Stephen Gildea for the MIT style file off which this current version is based, and Paolo Gaudiano for porting the MIT style to one compatible with BU requirements.

Janusz Konrad

Professor

ECE Department

SOME EXTENSIONS OF KLEENE ALGEBRA AND THEIR APPLICATIONS

CHENG ZHANG

Boston University, College of Engineering, 2024

Major Professors: First M. Last, PhD
Professor of Electrical and Computer Engineering
Secondary appointment

First M. Last, PhD
Professor of Computer Science

ABSTRACT

Have you ever wondered why this is called an *abstract*? Weird thing is that its legal to cite the abstract of a dissertation alone, apart from the rest of the manuscript.

Contents

List of Tables

List of Figures

List of Abbreviations

As per BU library instructions, the list of abbreviations must be in alphabetical order by the **abbreviation**, not by the explanation, or it will be returned to you for re-ordering. **This comment must be removed in the final document.**

CAD	.....	Computer-Aided Design
CO	.....	Cytochrome Oxidase
DOG	.....	Difference Of Gaussian (distributions)
FWHM	.....	Full-Width at Half Maximum
LGN	.....	Lateral Geniculate Nucleus
ODC	.....	Ocular Dominance Column
PDF	.....	Probability Distribution Function
$\mathbb{R}^2$	.....	the Real plane

Chapter 1

Introduction

1.1 A Brief History Of Kleene Algebra

Our Contributions

1.2 Technical Background

1.2.1 Kleene Algebra and Homomorphisms

A *Kleene Algebra* (KA) is an idempotent semiring with an iteration operator $(-)^*$. For a KA $\mathcal{K}$ and elements $p, q, r \in \mathcal{K}$ the complete list of axiom is listed below:

$$\begin{aligned}
p + 0 = 0 + p = p && \text{identity}\\
p + q = q + p && \text{commutativity}\\
(p + q) + r = p + (q + r) && \text{associativity}\\
p + p = p && \text{idempotency}\\
1p = p1 = p && \text{identity}\\
(pq)r = p(qr) && \text{associativity}\\
(p + q)r = pr + qr && \text{right distributivity}\\
r(p + q) = rp + rq && \text{left distributivity}\\
0p = p0 = 0 && \text{annihilation}\\
1 + p^*p = 1 + pp^* = p^* && \text{unfolding}\\
q + pr \leq r \Longrightarrow p^*q \leq r && \text{induction}\\
q + rp \leq r \Longrightarrow qp^* \leq r && \text{induction;}
\end{aligned}$$

the ordering $\leq$ in KA inherits the conventional ordering in idempotent semiring

$$p \leq q \Longleftrightarrow p + q = q.$$

Like every other algebraic structures, mapping between KA is an important subject of study. As we will show later, a lot of the known concepts in KA can be defined using properties of maps between KA. A *KA homomorphism* between two KA $\mathcal{K}$ and $\mathcal{K}'$ is a map $h : \mathcal{K} \to \mathcal{K}'$ that preserves all the KA operations, namely, for all

$p, q \in \mathcal{K}$ the following equations hold:

$$h(p+q) = h(p) + h(q)$$
$$h(p \cdot q) = h(p) \cdot h(q)$$
$$h(p^*) = h(p)^*.$$

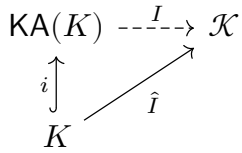
1.2.2 Free KA And Interpretations

In order to reason about logical properties like completeness about KA, it is important to define expressions and interpretations in Kleene algebra. A *KA alphabet* K is a finite set; we call the elements $p \in K$ *primitives*. Given an alphabet K, we can construct all the Kleene Algebra terms over K with the following grammar:

$$e = p \in K \mid 1 \mid 0 \mid e_1 + e_2 \mid e_1 \cdot e_2 \mid e^*.$$

And these terms forms the free KA:

Definition 1 (Free KA)**.** *The free KA* $\mathsf{KA}(K)$ *over an alphabet* K *is a KA s.t. for all KA* $\mathcal{K}$ *and a set map* $\hat{I} : K \to \mathcal{K}$*, there exists a unique KA homomorphism* $I : \mathsf{KA}(K) \to \mathcal{K}$*, where the following diagram commutes:*



where $i : K \to \mathsf{KA}(K)$ *is the inclusion map. Since there exists a bijection between set maps* $\hat{I} : K \to \mathcal{K}$ *and homomorphisms* $I : K \to \mathcal{K}$*, we will use* I *to denote both* $\hat{I}$ *and* I *for simplicity. We will also sometimes omit* K *and just write* KA *if* K *is irrelevant or can be inferred from context. We sometimes say* I *is* lifted *from the action on the primitives* $\hat{I}$*.*

The free KA over K can be constructed as all the KA terms over K modulo provable equalities in KA. The operations in free KA is purely syntactic, for example, the

addition of two terms $e_1, e_2 \in \mathsf{KA}(K)$ will simply yield the term $e_1 + e_2$. Furthermore, the homomorphism $I : \mathsf{KA}(K) \to \mathcal{K}$ is inductively generated from $\hat{I} : K \to \mathcal{K}$ as follows:

$$
\begin{aligned}
&I : \mathsf{KA}(K) \to \mathcal{K} \\
&I(p) \triangleq \hat{I}(p) && p \in K \\
&I(e_1 + e_2) \triangleq I(e_1) + I(e_2) \\
&I(e_1 \cdot e_2) \triangleq I(e_1) \cdot I(e_2) \\
&I(t^*) \triangleq I(t)^*
\end{aligned}
$$

The uniqueness of I can be derived by unfolding the definition of homomorphisms. Homomorphisms from free KA is called *KA interpretations*. By definition of the free KA, a KA interpretation is uniquely determined by the action on the primitives, i.e. for two interpretations $I, I' : \mathsf{KA}(K) \to \mathcal{K}$:

$$\forall p \in K, I(p) = I'(p) \iff \forall e \in \mathsf{KA}(K), I(e) = I'(e).$$

1.2.3 Complete Models

One important property for interpretation is completeness. We say an interpretation I is *complete* when the following equivalence holds:

$$\forall e_1, e_2 \in \mathsf{KA}(K), e_1 = e_2 \iff I(e_1) = I(e_2).$$

If we consider I as a desirable semantics for $\mathsf{KA}(K)$, then this means that all the semantical equalities can be derived using just the theory, a very desirable quality of a proof system.

One of such complete interpretation, as we have mentioned before, is the language interpretation. Kozen has showed that the powerset of any monoid generates a Kleene

Algebra [Koz02], and the language KA over alphabet K is generated by the free monoid over K, where we denote the identity as ϵ. More explicitly, the language KA over K can be defined as follows:

- The carrier set is all the languages (sets of words) over the alphabet K.
- The additive identity 0 is the empty set.
- The multiplicative identity 1 is the singleton set with the empty string $\{\epsilon\}$.
- The addition operation is set union.
- The multiplication operation is element-wise concatenation:

$$l_1 \cdot l_2 \triangleq \{w_1 w_2 \mid w_1 \in l_1, w_2 \in l_2\}.$$

- The star operation is the closure under finite concatenation:

$$l^* \triangleq \bigcup_{i \in \mathbb{N}} l^i \text{ where } l^0 \triangleq \epsilon, l^{(j+1)} \triangleq l^j \cdot l.$$

The language interpretation of KA is defined by lifting the following action on primitives p:

$$L(p) \triangleq \{p\}.$$

The completeness result of the language interpretation is a seminal result in Kleene algebra, proven independently by Kozen and Korb [Koz94, Kro91], and the proof was later improved several times [Koz01, Sil10, KS20].

1.2.4 Constructing Models

Upper-triangular matrix model enjoys nice properties that will be crucial in later development. Given a KA $\mathcal{K}$, the square upper-triangular matrices over $\mathcal{K}$ of size n, denoted as $M_n(\mathcal{K})$, forms a Kleene Algebra, with matrix addition, matrix multipli-

cation, and a star operation inductively defined as follows [Koz94]:

$$\begin{bmatrix} A & B \\ 0 & D \end{bmatrix}^* \triangleq \begin{bmatrix} A^* & A^*BD^* \\ 0 & D^* \end{bmatrix},$$

where A, D are square block matrices, and B is a block matrix.

Corollary 1. *Projections of the diagonal element of upper-triangular matrices are homomorphisms:*

$$\pi_{n,n} : M_m(\mathcal{K}) \to \mathcal{K} \text{ where } n \leq m.$$

Proof. By unfolding the definition □

Corollary 2. *Given an interpretation into a matrix model $I : \mathsf{KA}(X) \to M_m(\mathcal{K})$ and a Kleene subalgebra $\mathcal{K}' \subseteq \mathcal{K}$, any diagonal projection $\pi_{n,n}$ satisfy the following equivalences:*

$$\forall a \in X, \pi_{n,n}(I(a)) \subseteq \mathcal{K}' \iff \forall e \in \mathsf{KA}(X), \pi_{n,n}(I(e)) \subseteq \mathcal{K}'.$$

Proof. Direct consequence of Corollary 1. □

Another important class of models are models bounded by a single element:

Theorem 1. *Given a KA $\mathcal{K}$, and an element $p \in \mathcal{K}$, if $p \geq 1$ and $p \cdot p \leq p$, then all the element in $\mathcal{K}$ that is smaller than p forms a Kleene Algebra.*

Proof. We need to show that $\{q \mid q \leq p\}$ is closed under all operation of KA. We will show the star case as an example. Given an element $q \leq p$, we need to show $q^* \leq p$. By induction rule

$$qp \leq pp \leq p \Longrightarrow q^*p \leq p \Longrightarrow q^* \leq q^*p \leq p.$$ □

We denote the Kleene algebra formed by all the elements in $\mathcal{K}$ that is less than p as $\mathcal{K}_p$.

1.2.5 KAT

A classical extension of KA is Kleene Algebra with Tests (KAT) [KS97], which is a two-sorted algebra $(\mathcal{B}, \mathcal{K})$, s.t. $\mathcal{B}$ is a boolean algebra; $\mathcal{K}$ is a Kleene algebra; and $\mathcal{B}$

is a subalgebra of $\mathcal{K}$ where the disjunction and conjunction operator in $\mathcal{B}$ respectively coincide with the addition and multiplication in $\mathcal{K}$.

A *KAT alphabet* consists of two disjoint finite set K, B; the elements in K are called *primitive actions* and the elements in B are called *primitive tests*. Given a KAT alphabet K, B, we can define KAT terms as follows:

$$t = p \in K \mid b \in B \mid \overline{t_b} \mid 1 \mid 0 \mid t_1 + t_2 \mid t_1 \cdot t_2 \mid t^*,$$

where t_b do not contain any primitive actions. Notice boolean terms like $t_1 \vee t_2$ are not presented here, this is because the disjunction operator coincide with addition, hence by definition $t_1 \vee t_2$ is the same as $t_1 + t_2$.

Chapter 2

KA with Atomic Commutativity

2.1 Free KA with Atomic Commutativity

It is common to extend Kleene Algebra with additional equations to enrich the theory [DKPP19, KM14, PRW21]. In this paper we will consider atomic commutativity hypotheses, where the equations in the hypotheses are of the form $pq = qp$ with p and q being primitives.

A *commutable set* $(X, \sim)$ is a set with a reflexive symmetric relation $\sim\colon X \times X$ called *commuting relation*, we typically omit $\sim$ and just denote the commutable set as X. In this paper we only consider finite commutable sets.

We say a commutable set X is *discrete* if the relation $\sim$ is the identity relation. A homomorphism $h : X \to Y$ between two commutable set X and Y is a function that preserves the commuting relation:

$$x_1 \sim x_2 \Longrightarrow h(x_1) \sim h(x_2).$$

The carrier of a commutable set X can be considered as a discrete commutable set, and we denote this discrete commutable set as $X_{\nsim}$. There is a canonical homomorphism:

$$[-]_\sim : X_{\nsim} \to X$$
$$[x]_\sim \triangleq x.$$

We can construct the free KA a commutable set X by taking all the KA terms

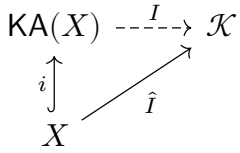
over X modulo the equalities provable from KA axioms plus the following equations $\{pq = qp \mid p \sim q\}$. Intuitively, the commuting relation of X specifies the atomic commutativity hypotheses in $\mathsf{KA}(X)$. Since the free KA over a set is just a free KA over discrete commutable set, we abuse the notation to denote the free KA over a commutable set X as $\mathsf{KA}(X)$.

Notice all Kleene Algebra form a commutable set, with the commuting relation defined as follows:

$$e_1 \sim e_2 \iff e_1 \cdot e_2 = e_2 \cdot e_1.$$

We can show that the free KA over commutable set enjoys similar universal property as free KA over set. We first prove the universal property without the uniqueness requirement:

Theorem 2. *For all commutable set X, a KA $\mathcal{K}$, and a commutable set homomorphism $\hat{I} : X \to \mathcal{K}$, then there is a KA interpretation $I : \mathsf{KA}(X) \to \mathcal{K}$, s.t. the following diagram commutes:*



Proof. Given the function $\hat{I}$, we can apply the standard technique to generate the homomorphism I by induction on the input:

$$\begin{aligned}
I(a) &\triangleq \hat{I}(a) && a \in X \\
I(1) &\triangleq 1_{\mathcal{K}} \\
I(0) &\triangleq 0_{\mathcal{K}} \\
I(e_1 + e_2) &\triangleq I(e_1) + I(e_2) \\
I(e_1 \cdot e_2) &\triangleq I(e_1) \cdot I(e_2) \\
I(e^*) &\triangleq I(e)^*
\end{aligned}$$

such a homomorphism exists, and makes the diagram commute:

$$I(a) = \hat{I}(a), \forall a \in X.$$

□

To prove uniqueness, we will prove a stronger theorem first.

Theorem 3. *Given two interpretation* $I, I' : \mathsf{KA}(X) \to \mathcal{K}$*,*

$$I(e) \geq I'(e) \iff \forall a \in X, I(a) \geq I'(a),$$

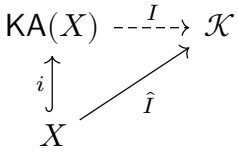
this result implies $I(e) = I'(e) \iff \forall a \in X, I(a) = I'(a)$*.*

Proof. By induction on the structure of e, and all KA operations preserve order. We show the star case as example: assume $I(e) \geq I'(e)$, we need to show $I(e^*) \geq I'(e^*)$. Since I is a homomorphism, and star preserves order:

$$I(e^*) = (I(e))^* \geq (I'(e))^* = I'(e^*).$$

□

Corollary 3 (Universal Property)**.** *For all commutable set* X*, a KA* $\mathcal{K}$*, and a commutable set homomorphism* $\hat{I} : X \to \mathcal{K}$*, then there is a unique KA interpretation* $I : \mathsf{KA}(X) \to \mathcal{K}$*, s.t. the following diagram commutes:*



Proof. By Theorem 2, I exists. By Theorem 3, if there exists another interpretation I' that makes the diagram commute, then

$$I(a) = I'(a), \forall a \in X \implies I(e) = I'(e).$$

□

As usual, we will use the notation I for both I and $\hat{I}$.

The words over a commutable set X are monoid terms modulo monoid equations plus the commutativity axioms $\{ab = ba \mid a \sim b\}$. We still use ϵ as the identity of the monoid and call it the empty word; and we use the same notation $\mathsf{Wrd}(X)$ for all the words over X. The language model over a commutable set X is the powerset of all words over X, with operation defined by Kozen [Koz02], denoted as $\mathcal{L}_X$. The language interpretation is generated by the same action on primitives as in Kleene Algebra:

$$L : X \to \mathcal{L}_X$$
$$L(a) = \{a\}$$

Notation In the rest of the article, notations $\mathsf{KA}(X)$, $\mathsf{Wrd}(X)$, and $\mathcal{L}_X$ always refers to the commutative variant, where X is a commutable set. When we are referring to the non-commutative KA (word, language model, etc.), we will consider them as the KA (word, language model, etc.) over a discrete commutable set. As we have mentioned before, $2 \triangleq \{0, 1\}$ denotes the unique KA that only contains two distinct identities; this KA is also the free KA generated by the empty set $\mathsf{KA}(\emptyset)$. Finally, when given a finite set of terms $S \subseteq \mathsf{KA}(X)$, we will sometimes use S to denote the sum of all its elements $(\sum_{e \in S} e) \in \mathsf{KA}(e)$.

2.2 Word Inhabitant Problem

Given a commutable set, we allow a word in $\mathcal{L}_X$ to be implicitly coerced into $\mathsf{KA}(X)$, where we pick the multiplication operator in $\mathsf{KA}(X)$ as the monoidal multiplication.

Then given a word $w \in \mathcal{L}_X$ and a KA expression $e \in \mathsf{KA}(X)$, the word inhabitant problem is the following inequality:

$$w \leq e.$$

The problem is complete with language interpretation when:

$$w \in L(e) \iff w \leq e.$$

We will show that the word inhabitance problem is complete and decidable in Kleene Algebra with atomic commutativity hypotheses.

The core technique of this section is to construct a sound empty word predicate $E : \mathsf{KA}(X) \to 2$ and derivative operation $\delta_a : \mathsf{KA}(X) \to \mathsf{KA}(X)$.

2.2.1 Empty Word Predicate

In this section we will prove a stronger result than the soundness of empty word predicate:

$$\forall e \in \mathsf{KA}(X), e = E(e) + e',$$

where $E : \mathsf{KA}(X) \to 2$ is the empty word predicate on the free KA over any commutable set X, and $\epsilon \notin L(e')$. This result is obtained by decomposing using the following matrix model.

Theorem 4. *For any Kleene Algebra $\mathcal{K}$, matrix of the following shapes forms a Kleene Algebra:*

$$D_E(\mathcal{K}) \triangleq \{\begin{bmatrix} p & q \\ 0 & p+q \end{bmatrix} \mid p, q \in \mathcal{K}\}.$$

Proof. We will only need to show that matrix of this shape is closed under all the KA operations.

The identities and addition are easy to verify. So we will only focus on verifying the closure under multiplication and star operation.

The multiplication case:

$$\begin{bmatrix} p_1 & q_1 \\ 0 & p_1+q_1 \end{bmatrix} \begin{bmatrix} p_2 & q_2 \\ 0 & p_2+q_2 \end{bmatrix} = \begin{bmatrix} p_1p_2 & p_1q_2 + q_1(p_2+q_2) \\ 0 & (p_1+q_1)(p_2+q_2) \end{bmatrix}.$$

Since $p_1p_2 + p_1q_2 + q_1(p_2+q_2) = (p_1+q_1)(p_2+q_2)$, these matrices are closed under multiplication.

The star case:

$$\begin{bmatrix} p & q \\ 0 & p+q \end{bmatrix}^* = \begin{bmatrix} p^* & p^*q(p+q)^* \\ 0 & (p+q)^* \end{bmatrix}.$$

With a standard theorem of KA $(p+q)^* = p^*(qp^*)^*$, we are able to derive the closure under star operation:

$$\begin{aligned} p^* + p^*q(p+q)^* &= p^* + p^*qp^*(qp^*)^* \\ &= p^*(1 + qp^*(qp^*)^*) \\ &= p^*(qp^*)^* = (p+q)^* \end{aligned}$$

□

Given any commutable set X, consider the following matrix:

$$D_E(\mathsf{KA}(X)) \ni u_E \triangleq \begin{bmatrix} 1 & XX^* \\ 0 & X^* \end{bmatrix},$$

where X is a shorthand for the expression $(\sum_{x \in X} x)$. By simply unfolding the definition, we can verify that $u_E \cdot u_E = u_E$ and $u_E \geq 1$. Therefore, all the matrices less than u_E in $D_E(\mathsf{KA}(X))$ forms a Kleene Algebra. We denote this Kleene Algebra as $D_E(\mathsf{KA}(X))_{u_E}$.

In order to decompose an arbitrary expression, we will define an interpretation into $D_E(\mathsf{KA}(X))_{u_E}$ by lifting the following actions

$$\begin{aligned} I_E &: \mathsf{KA}(X) \to D_E(\mathsf{KA}(X))_{u_E} \\ I_E(a) &\triangleq \begin{bmatrix} 0 & a \\ 0 & a \end{bmatrix}. \end{aligned}$$

Because the projection $\pi_{2,2}$ is a homomorphism, then $\pi_{2,2} \circ I_E$ is an interpretation. Recall that interpretation is uniquely determined by the action on the primitives, and

$$\pi_{2,2} \circ I_E(a) = a, \forall a \in X.$$

Therefore, for all term in $e \in \mathsf{KA}(X)$, the $2,2$ component of $I_E(e)$ is exactly e itself:

$$\pi_{2,2} \circ I_E(e) = e.$$

Then we define the empty word predicate as follows:

$$E(e) \triangleq \pi_{1,1}(I_E(e)), \quad e' \triangleq \pi_{1,2}(I_E(e)).$$

By Corollary 2, and $\pi_{1,1}(I_E(a)) = 0 \in 2$ for all primitives a,

$$\forall e \in \mathsf{KA}(X), E(e) = \pi_{1,1}(I_E(e)) \in 2 \subseteq \mathsf{KA}(X).$$

Therefore, we can treat E as a homomorphism of the type $\mathsf{KA}(X) \to 2$.

Corollary 4 (empty word decomposition)**.** *All expression $e \in \mathsf{KA}(X)$ over a commutable set X can be decomposed in the following way:*

$$e = E(e) + e' \text{ where } \epsilon \notin L(e').$$

Proof. Recall that

$$\begin{bmatrix} E(e) & e' \\ 0 & e \end{bmatrix} \triangleq I_E(e).$$

Since $I_E(e) \in D_E(\mathsf{KA}(X))$, we have

$$e = E(e) + e'.$$

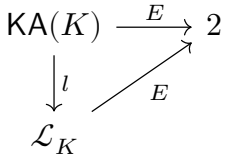
Furthermore, since elements in $D_E(\mathsf{KA}(X))_{u_E}$ is bounded by u_E,

$$e' = \pi_{2,2}(I_E(e)) \leq XX^*.$$

Because $\epsilon \notin L(XX^*)$, and l is a homomorphism, we conclude $\epsilon \notin L(e) \subseteq L(XX^*)$. □

Corollary 5 (Soundness Of Empty Word Property)**.** *Let $E : \mathcal{L}_X \to 2$ the empty word predicate on the language over a commutable set $E(l) = \epsilon \in l$, then the following*

diagram commute



Proof. We only need to prove that for all $e \in \mathsf{KA}(X)$,

$$E(e) = 1 \iff \epsilon \in l.$$

We show this by case analysis on $E(e)$:

- If $E(e) = 1$, then

$$L(e) = L(E(e)) \cup L(e') = \{\epsilon\} \cup L(e') \ni \epsilon.$$

- If $E(e) = 0$, recall that $\epsilon \notin L(e')$,

$$L(e) = L(E(e)) \cup L(e') = L(e') \not\ni \epsilon.$$

□

2.2.2 Derivative

Similar to the last section, the derivative operation will also be defined by a decomposition: for all $e \in \mathsf{KA}(X)$ and $a \in X$,

$$e = a \cdot \delta_a(e) + \rho_a(e),$$

where the language interpretation for $a \cdot \delta_a(e)$ and $\rho_a(e)$ are disjoint. This result will imply the soundness of derivative.

Theorem 5. *Given a KA $\mathcal{K}$ and an element $t \in \mathcal{K}$, the following matrices form a KA:*

$$D_t(\mathcal{K}) = \{ \begin{bmatrix} a & b & c \\ 0 & d & 0 \\ 0 & 0 & d \end{bmatrix} \mid d = a + b + tc, at = ta \}$$

Proof. We need to show that these matrices are closed under KA operations. The closure under identities and addition are trivial, we only show the multiplication case and the star case.

The multiplication case:

$$\begin{aligned}&\begin{bmatrix} p_1 & q_1 & r_1 \\ 0 & s_1 & 0 \\ 0 & 0 & s_1 \end{bmatrix} \begin{bmatrix} p_2 & q_2 & r_2 \\ 0 & s_2 & 0 \\ 0 & 0 & s_2 \end{bmatrix} \\ = &\begin{bmatrix} p_1p_2 & p_1q_2 + q_1s_2 & p_1r_2 + r_1s_2 \\ 0 & s_1s_2 & 0 \\ 0 & 0 & s_1s_2 \end{bmatrix}\end{aligned}$$

We verify that the equation is preserved:

$$\begin{aligned} s_1s_2 &= (p_1 + q_1 + pr_1) \cdot s_2 \\ &= p_1s_2 + q_1s_2 + pr_1s_2 \\ &= p_1(p_2 + q_2 + pr_2) + q_1s_2 + pr_1s_2 \\ &= p_1p_2 + (p_1q_2 + q_1s_2) + p_1pr_2 + pr_1s_2 \\ &= p_1p_2 + (p_1q_2 + q_1s_2) + p(p_1r_2 + r_1s_2) \end{aligned}$$

The last step uses the commutativity of t and p_1. Then we verify the commutativity condition:

$$(p_1p_2)t = p_1tp_2 = t(p_1p_2).$$

Hence, $D_t(\mathcal{K})$ is closed under multiplication.

The star case:

$$\begin{bmatrix} p & q & r \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix}^* = \begin{bmatrix} p^* & p^*qs^* & p^*rs^* \\ 0 & s^* & 0 \\ 0 & 0 & s^* \end{bmatrix}$$

the equation is preserved:

$$\begin{aligned} s^* &= (p + q + tc)^* \\ &= p^*((q + pc)p^*)^* \\ &= p^*(1 + (q + pc)p^*((q + pc)p^*)^*) \\ &= p^*(1 + (q + pc)s^*) \\ &= p^* + p^*qs^* + p^*trs^* \\ &= p^* + p^*qs^* + tp^*rs^* \end{aligned}$$

The last line is by standard KA theorem:

$$pt = tp \Longrightarrow p^*t = tp^*.$$

The commutativity condition $p^*t = tp^*$ is also implied by the above theorem. Therefore, $D_t(\mathcal{K})$ is closes under star operations. □

Given a commutable set X, and an element $a \in X$, we can partition the rest of the elements in X by whether they commute with a:

$$X_{\sim a} \triangleq \{b \mid b \sim a, b \neq a\}, \quad X_{\nsim a} = \{b \mid b \nsim a\}.$$

Since a commutes with every element of $X_{\sim a}$, a commutes with $X_{\sim a}$: $X_{\sim a} \cdot a = a \cdot X_{\sim a}$, then by standard theorem of KA:

$$X^*_{\sim a} \cdot a = a \cdot X^*_{\sim a}.$$

Consider the following matrix:

$$D_a(\mathsf{KA}(X)) \ni u_a = \begin{bmatrix} X^*_{\sim a} & X^*_{\sim a} X_{\nsim a} X^* & X^* \\ 0 & X^* & 0 \\ 0 & 0 & X^* \end{bmatrix}.$$

It is easy to verify that $u_a \geq 1$ and $u_a \cdot u_a \leq u_a$. Therefore, the elements under u_a forms a KA: $D_a(\mathsf{KA}(X))_{u_a}$. The purpose of model $D_a(\mathsf{KA}(X))_{u_a}$ is clear when we look at the language interpretation for each of the component, let

$$\begin{bmatrix} p & q & r \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} \in D_a(\mathsf{KA}(X))_{u_a}$$

then

- $L(p) \leq L(X^*_{\sim a})$ contains only words with symbols that commutes with primitive a, but is not a.

- $L(q) \leq L(X^*_{\sim a} X_{\nsim a} X^*)$ contains words that starts with arbitrary number of primitives that commutes with a, then a primitive that does not commute with a, followed by arbitrary primitives.

Both $L(p)$ and $L(q)$ do not contain words of the form $a \cdot w$ for any word $w \in \mathsf{Wrd}(X)$; by the property of $D_a(\mathsf{KA}(X))$:

$$L(s) = L(p) + L(q) + a \cdot L(r).$$

Thus, $L(r)$ will be the language derivative of $L(s)$ with respect to primitive a,

To apply this decomposition on an arbitrary expression, we define an interpretation by lifting the following action on primitives:

$$I_a : X \to D_a(\mathsf{KA}(X))_{u_a}$$

$$I_a(b) \triangleq \begin{cases} \begin{bmatrix} b & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & b \end{bmatrix} & b \in X_{\sim a} \\ \begin{bmatrix} 0 & b & 0 \\ 0 & b & 0 \\ 0 & 0 & b \end{bmatrix} & b \in X_{\nsim a} \\ \begin{bmatrix} 0 & 0 & 1 \\ 0 & b & 0 \\ 0 & 0 & b \end{bmatrix} & b = a \end{cases}$$

Again, since $\pi_{3,3}$ and $\pi_{2,2}$ are homomorphisms, $\pi_{3,3} \circ I_a$ and $\pi_{2,2} \circ I_a$ are interpretations. Because interpretations are uniquely determined by the action on the

primitives, and

$$\forall b \in X, \pi_{3,3} \circ I_a(b) = \pi_{2,2} \circ I_a(b) = b,$$

then $\pi_{3,3} \circ I_a$ and $\pi_{2,2} \circ I_a$ are both identity homomorphisms. This means the $2,2$ and $3,3$ component of $I_a(e)$ are exactly e for all $e \in \mathsf{KA}(X)$. Let

$$\begin{bmatrix} p & q & r \\ 0 & e & 0 \\ 0 & 0 & e \end{bmatrix} \triangleq I_a(e) \in D_a(\mathsf{KA}(X))_{u_a}.$$

We can define the derivative δ_a and residual ρ_a as follows:

$$\delta_a(e) \triangleq r, \quad \rho_a(e) \triangleq p + q.$$

Then the following corollary can be derived simply from the definition of $D_a(\mathsf{KA}(X))_{u_a}$.

Corollary 6 (decomposition)**.** *For all expressions $e \in \mathsf{KA}(X)$, primitives $a \in X$, and word $w \in \mathcal{L}_X$,*

$$e = \rho_a(e) + a \cdot \delta_a(e) \text{ and } a \cdot w \notin L(\rho_a(e)).$$

Theorem 6 (Soundness Property)**.** *For a primitive a in a commutable set X, let the derivative on language δ_a defined as $\delta_a(l) \triangleq \{s \mid a \cdot w \in l\}$, the following diagram commute:*

$$\begin{array}{ccc} \mathsf{KA}(X) & \xrightarrow{\delta_a} & \mathsf{KA}(X) \\ \downarrow{\scriptstyle L} & & \downarrow{\scriptstyle L} \\ \mathcal{L}_X & \xrightarrow{\delta_a} & \mathcal{L}_X \end{array}$$

Proof. Given any word $w \in \mathcal{L}_X$ and in $e \in \mathsf{KA}(X)$:

$$\begin{aligned} & s \in \delta_a(L(e)) \\ \iff & a \cdot w \in L(e) && \text{by definition of language } \delta_a \\ \iff & a \cdot w \in L(\rho_a(e) + a \cdot \delta_a(e)) && \text{Corollary 6} \\ \iff & a \cdot w \in L(a \cdot \delta_a(e)) && a \cdot w \notin L(\rho_a(e)) \\ \iff & s \in L(\delta_a(e)). \end{aligned}$$

Thus, for all $e \in \mathsf{KA}(X)$, $\delta_a(L(e)) = L(\delta_a(e))$, we have reached our conclusion. □

Finally, we prove a Galois connection that the derivative is expected to satisfy.

Lemma 1 (Basic Algebraic Properties)**.** *Following basic algebraic properties are true, for all primitive a and expressions e, e':*

$$\begin{aligned}
\delta_a(ae) &= e \\
\rho_a(ae) &= 0 \\
\delta_a(\rho_a(e)) &= 0 \\
e \geq e' &\Longrightarrow \delta_a(e) \geq \delta_a(e') \\
e \geq e' &\Longrightarrow \rho_a(e) \geq \rho_a(e')
\end{aligned}$$

Proof. We first compute $I_a(ae)$:

$$I_a(ae) = I_a(a) \cdot I_a(e) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & a & 0 \\ 0 & 0 & a \end{bmatrix} \begin{bmatrix} p & q & r \\ 0 & e & 0 \\ 0 & 0 & e \end{bmatrix},$$

for some expressions $p, q, r \in \mathsf{KA}(X)$. Then

$$I_a(ae) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & a & 0 \\ 0 & 0 & a \end{bmatrix} \begin{bmatrix} p & q & r \\ 0 & e & 0 \\ 0 & 0 & e \end{bmatrix} = \begin{bmatrix} 0 & 0 & e \\ 0 & ae & 0 \\ 0 & 0 & ae \end{bmatrix}.$$

Therefore, we obtain the conclusion $\delta_a(ae) = e$ and $\rho_a(ae) = 0$.

Notice that $\rho(e) \leq X^*_{\sim a} + X^*_{\sim a} X_{\nsim a} X^*$, therefore

$$\begin{aligned}
I_a(\rho(e)) &\leq I_a(X^*_{\sim a} + X^*_{\sim a} X_{\nsim a} X^*) \\
&= \begin{bmatrix} X^*_{\sim a} & X^*_{\sim a} X_{\nsim a} X^* & 0 \\ 0 & X^*_{\sim a} + X^*_{\sim a} X_{\nsim a} X^* & 0 \\ 0 & 0 & X^*_{\sim a} + X^*_{\sim a} X_{\nsim a} X^* \end{bmatrix}
\end{aligned}$$

Therefore $I_a(\rho(e)) \leq 0$, and since 0 is the smallest element, We obtain the conclusion $I_a(\rho(e)) = 0$.

The monotonicity can be derived from the monotonicity of I_a. When $e \geq e'$, we have $I_a(e) \geq I_a(e')$. Recall that the ordering on matrices are component order, since $\delta_a(e)$ and $\rho_a(e)$ are either component of $I_a(e)$ or the sum of components of $I_a(e)$, therefore $\delta_a(e) \geq \delta_a(e')$ and $\rho_a(e) \geq \rho_a(e')$. □

Theorem 7 (Galois Connection)**.** *Given a commutative set X, for all expression*

$e, e' \in \mathsf{KA}(X)$ *and primitive* $a \in X$,

$$ae \leq e' \iff e \leq \delta_a(e').$$

Proof. We first show $ae \leq e' \iff e \leq \delta_a(e')$: $\implies$ direction can be proved by applying δ_a to both sides:

$$ae \leq e' \implies \delta_a(ae) \leq \delta_a(e') \implies e \leq \delta_a(e').$$

$\impliedby$ direction proven by multiplying a on both sides:

$$e \leq \delta_a(e') \implies ae \leq a\delta_a(e') \implies ae \leq a\delta_a(e') \leq e'.$$

□

2.2.3 Decidability And Completeness

In this section, we prove the completeness and decidability of the word inhabitance problem, by explicitly define an algorithm to check for word inhabitance.

Theorem 8 (Decidability and Completeness)**.** *Given a word* $w \in \mathsf{Wrd}(X)$ *and an expression* $e \in \mathsf{KA}(X)$*, we can define the following algorithm to test for inhabitants:*

$$\begin{aligned} &i : \mathsf{Wrd}(X) \times \mathsf{KA}(X) \to 2 \\ &i(\epsilon, e) \triangleq E(e) \\ &i(a \cdot w, e) \triangleq i(w, \delta_a(e)) \end{aligned}$$

Such an algorithm will always terminate, and it is sound:

$$w \in L(e) \iff i(w, e) = 1 \iff w \leq e.$$

Proof. The algorithm i will terminate because both δ_a and E can be computed by computing the interpretation I_a and I_E.

We first show $w \in L(e) \iff i(w, e)$ by induction on w.

- If $w = \epsilon$, then $\epsilon \in L(e) \iff E(e) = 1$ by soundness of E.

- If $w = a \cdot w'$ then:

$$\begin{aligned} a \cdot w' \in L(e) &\iff w' \in \delta_a(L(e)) && \text{definition} \\ &\iff w' \in L(\delta_a(e)) && \text{soundness of } \delta_a \\ &\iff i(w, \delta_a(e)) && \text{induction hypothesis} \end{aligned}$$

We then show $i(w, e) = 1 \iff w \leq e$ by induction on w.

- If $w = \epsilon$, then $i(\epsilon, e) = E(e)$. When $E(e) = 1$, then $1 = E(e) \leq e$; When $E(e) = 0$, then $1 \not\leq e$, because $1 \leq e$ is not true in the language interpretation.
- If $w = a \cdot w'$ then:

$$\begin{aligned} a \cdot w' \leq e &\iff w' \leq \delta_a(e) && \text{Theorem 7} \\ &\iff i(w, \delta_a(e)) && \text{induction hypothesis} \end{aligned}$$

□

2.2.4 Fundamental Theorem

Fundamental theorem is an important soundness condition for the definition of derivative and empty word predicate, it also exhibits a strong connection between KA and automata [Sil10, KS20]. Because of the significance of the fundamental theorem, we decide to prove it for KA with atomic commutativity, despite it is not used in the rest of the paper.

In order to show the fundamental theorem for KA with atomic commutativity, we will establish the relation between derivative and empty word predicate in KA with their counterparts in KA with commutativity, and with this relation, we can show that fundamental theorem KA implies the fundamental theorem in KA with commutativity

Recall that for all commutable set X, we can construct a discrete commutable set $X_\sim$ by replacing the commuting relation in X with the identity relation. Notice

that $\mathsf{KA}(X_\asymp)$ is a free KA, and by **??**, derivative and empty word predicate is unique on free KAs. Since our definition of E and δ_a are sound, therefore they are exactly the conventional E and δ_a when applied to a term in the free KA. Therefore, the fundamental theorem holds for $\mathsf{KA}(X_\asymp)$:

$$\forall e_\asymp \in \mathsf{KA}(X_\asymp), e_\asymp = E(e_\asymp) + \sum_{a \in X} a \cdot \delta_a(e_\asymp).$$

Finally, there is a canonical KA homomorphism from $\mathsf{KA}(X_\asymp)$ to $\mathsf{KA}(X)$, by lifting the following action on the primitives:

$$[a]_\sim \triangleq a.$$

This KA homomorphism imposes the commutativity of X to the input expression, and this homomorphism is surjective.

Lemma 2. *Consider a Kleene Algebra $\mathcal{K}$ and an element $t \in \mathcal{K}$, there is a homomorphism:*

$$h : D_t(\mathcal{K}) \to M_2(\mathcal{K})$$
$$h(\begin{bmatrix} p & q & r \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix}) = \begin{bmatrix} p & r \\ 0 & s \end{bmatrix}$$

Proof. Perseverance of identities and addition is trivial, we will only check for perseverance of multiplication and star

The multiplication case:

$$\begin{aligned} & h(\begin{bmatrix} p_1 & q_1 & r_1 \\ 0 & s_1 & 0 \\ 0 & 0 & s_1 \end{bmatrix} \begin{bmatrix} p_2 & q_2 & r_2 \\ 0 & s_2 & 0 \\ 0 & 0 & s_2 \end{bmatrix}) \\ = & h(\begin{bmatrix} p_1p_2 & p_1q_2 + q_1s_2 & p_1r_2 + r_1s_2 \\ 0 & s_1s_2 & 0 \\ 0 & 0 & s_1s_2 \end{bmatrix}) \\ = & \begin{bmatrix} p_1p_2 & p_1r_2 + r_1s_2 \\ 0 & s_1s_2 \end{bmatrix} \\ = & h(\begin{bmatrix} p_1 & q_1 & r_1 \\ 0 & s_1 & 0 \\ 0 & 0 & s_1 \end{bmatrix}) \cdot h(\begin{bmatrix} p_2 & q_2 & r_2 \\ 0 & s_2 & 0 \\ 0 & 0 & s_2 \end{bmatrix}). \end{aligned}$$

The star case:

$$h(\begin{bmatrix} p & q & r \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix}^*) = h(\begin{bmatrix} p^* & p^*qs^* & p^*rs^* \\ 0 & s^* & 0 \\ 0 & 0 & s^* \end{bmatrix})$$
$$= \begin{bmatrix} p^* & p^*rs^* \\ 0 & s^* \end{bmatrix} = h(\begin{bmatrix} p & q & r \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix})^*.$$

□

Since the derivative $\delta_a(e)$ is defined as the $1,3$ component of $I_a(e)$, after we apply above homomorphism h to $I_a(e)$, the derivative $\delta_a(e)$ becomes the $1,2$ component of the matrix:

$$\forall e \in \mathsf{KA}(X), \delta_a(e) = \pi_{1,2}(h(I_a(e))).$$

Lemma 3. *Let X be a commutable set, for all non-commutativity expressions $e_\asymp \in \mathsf{KA}(X_\asymp)$:*

$$E([e_\asymp]_\sim) = [E(e_\asymp)]_\sim, \quad \delta_a([e_\asymp]_\sim) \geq [\delta_a(e_\asymp)]_\sim.$$

Proof. Consider the following interpretations

$$I_E \circ [-]_\sim \text{ and } [-]_\sim \circ I_E : \mathsf{KA}(X_\asymp) \to \mathsf{KA}(X).$$

Their actions coincide on the primitives:

$$\forall a \in X_\asymp, I_E([a]_\sim) = \begin{bmatrix} 0 & a \\ 0 & a \end{bmatrix} = [E(a)]_\sim.$$

Therefore, by Theorem 3,

$$\forall e_\asymp \in \mathsf{KA}(X_\asymp), I_E([e_\asymp]_\sim) = [I_E(e_\asymp)]_\sim.$$

Since E is a component of I_E,

$$E([e_\asymp]_\sim) = [E(e_\asymp)]_\sim.$$

The same can be done for derivatives. We consider the following interpretations:

$$h \circ I_a \circ [-]_\sim \text{ and } [-]_\sim \circ h \circ I_a : \mathsf{KA}(X_\asymp) \to \mathsf{KA}(X).$$

Given a primitive b,

- if $b = a$, then

$$\forall b \in X_{\nsim}, h(I_a([b]_\sim)) = \begin{bmatrix} 0 & 1 \\ 0 & b \end{bmatrix} = [h(I_a(b))]_\sim;$$

- if $b \neq a$, then

$$h \circ I_a([b]_\sim) = \begin{cases} \begin{bmatrix} 0 & 0 \\ 0 & b \end{bmatrix} & b \nsim a \text{ in } X \\ \begin{bmatrix} b & 0 \\ 0 & b \end{bmatrix} & b \sim a \text{ in } X \end{cases}$$

  both of which are greater than

$$[h(I_a(b))]_\sim = \begin{bmatrix} 0 & 0 \\ 0 & b \end{bmatrix} \quad \text{because } b \nsim a \text{ in } X_{\nsim}.$$

Therefore, for all primitive $b \in X_{\nsim}$, $h(I_a([b]_\sim)) \geq [h(I_a(b))]_\sim$. By Theorem 3

$$\forall e_{\nsim} \in \mathsf{KA}(X_{\nsim}), h \circ I_a([e_{\nsim}]_\sim) \geq [h \circ I_a(e_{\nsim})]_\sim.$$

Since δ_a is a component of $h \circ I_a$,

$$\delta_a([e_{\nsim}]_\sim) \geq [\delta_a(e_{\nsim})]_\sim. \qquad \square$$

The above lemma state that the derivative for commutative expression in $\mathsf{KA}(X)$ is always larger than their non-commutativity counterparts in $\mathsf{KA}(X_{\nsim})$. Since $\mathsf{KA}(X_{\nsim})$ is a free KA, we can easily derive the fundamental theorem for $\mathsf{KA}(X)$ using its connection with $\mathsf{KA}(X_{\nsim})$.

Theorem 9 (Fundamental Thoerem)**.** *For all $e \in \mathsf{KA}(X)$, the following equality holds:*

$$e = E(e) + \sum_{a \in X} a \cdot \delta_a(e)$$

Proof. We first show $e \geq E(e) + \sum_{a \in X} a \cdot \delta_a(e)$, which is a direct consequence of

decompositions in corollaries 4 and 6:

$$e \geq E(e),$$
$$e \geq a \cdot \delta_a(e), \forall a \in X.$$

We then show $e \leq E(e) + \sum_{a \in X} a \cdot \delta_a(e)$. Since $[-]_\sim$ is a surjective homomorphism, we consider $e_\asymp \in \mathsf{KA}(X_\asymp)$ s.t. $[e_\asymp]_\sim = e$. Because $\mathsf{KA}(X_\asymp)$ is a free KA, fundamental theorem holds for $e_\asymp$:

$$e_\asymp \leq E(e_\asymp) + \sum_{a \in X} a \cdot \delta_a(e_\asymp).$$

We can apply the homomorphism $[-]_\sim$ to both sides:

$$e \leq [E(e_\asymp)]_\sim + \sum_{a \in X} a \cdot [\delta_a(e_\asymp)]_\sim.$$

By Lemma 3,

$$[E(e_\asymp)]_\sim = E([e_\asymp]_\sim) = E(e),$$
$$[\delta_a(e_\asymp)]_\sim \leq \delta_a([e_\asymp]_\sim) = \delta_a(e).$$

Thus, obtain the desired inequality:

$$e \leq [E(e_\asymp)]_\sim + \sum_{a \in X} a \cdot [\delta_a(e_\asymp)]_\sim \leq E(e) + \sum_{a \in X} a \cdot \delta_a(e).$$

□

2.3 Undecidability

In this section, we will show the undecidability result for general Kleene Algebra equalities with atomic commutativity hypotheses. The undecidability result is obtained by using a proof to simulate the execution of a two-counter machine. From there, we can encode state reachability of terminating two-counter machines into an KA inequality. Enabling us to carry out a diagonal argument, similar to the proof of undecidability of halting problem.

2.3.1 Encoding Two-Counter Machines

Counter machine is a well-studied machine [Min61, Min67, Lam61], and it can simulate any Turing machine [Min67, Theorem 14.1-1] with just two counters. In this paper, we only consider two-counter machines. A two-counter machine $M \triangleq (S, \hat{s}, \iota)$ consists of a finite set of *state* S, a start state $\hat{s} \in S$, and each state is equipped with an instruction $\iota : S \to I_S$, where instructions I_S is defined as follows:

$$\begin{aligned} I_S \triangleq & \{\texttt{Inc}(s, q) \mid s \in \{1, 2\}, q \in Q\} \\ & \cup \{\texttt{Dec}(s, q) \mid s \in \{1, 2\}, q \in Q\} \\ & \cup \{\texttt{If}(s, q_1, q_2) \mid s \in \{1, 2\}, q_1, q_2 \in Q\} \\ & \cup \{\texttt{Halt}\}. \end{aligned}$$

Each instruction has a semantics, we define $S_\perp \triangleq S + \{\perp\}$:

$$\begin{aligned} [\![i]\!] &: \mathbb{N} \times \mathbb{N} \to S_\perp \times \mathbb{N} \times \mathbb{N} \\ [\![\texttt{Inc}(1, s)]\!](n, m) &\triangleq (s, n + 1, m) \\ [\![\texttt{Inc}(2, s)]\!](n, m) &\triangleq (s, n, m + 1) \\ [\![\texttt{Dec}(1, s)]\!](n, m) &\triangleq (s, \max(n - 1, 0), m) \\ [\![\texttt{Dec}(2, s)]\!](n, m) &\triangleq (s, n, \max(m - 1, 0)) \\ [\![\texttt{If}(1, q_1, q_2)]\!](n, m) &\triangleq \begin{cases} (s_1, n, m) & \text{if } n = 0 \\ (s_2, n, m) & \text{if } n \neq 0 \end{cases} \\ [\![\texttt{If}(2, s_1, s_2)]\!](n, m) &\triangleq \begin{cases} (s_1, n, m) & \text{if } m = 0 \\ (s_2, n, m) & \text{if } m \neq 0 \end{cases} \\ [\![\texttt{Halt}]\!](n, m) &\triangleq (\perp, n, m). \end{aligned}$$

From the semantics of the instruction, we can define a transition relation for any

machine M:

$$R_M \in (S \times \mathbb{N} \times \mathbb{N}) \times (S_\perp \times \mathbb{N} \times \mathbb{N})$$
$$R_M \triangleq \{((s, m, n), [\![\iota(s)]\!](m, n)) \mid s \in S; m, n \in \mathbb{N}\}.$$

Note that R_M is a functional relation, that is for all input $(s, m, n) \in (S \times \mathbb{N} \times \mathbb{N})$ there exists a unique element in $S_\perp \times \mathbb{N} \times \mathbb{N}$ relating to it. We call the elements in $S_\perp \times \mathbb{N} \times \mathbb{N}$ *configurations* of the machine M. Let R_M^* be the reflexive transition closure for R_M, and we write $c \to^* c'$ if $(c, c') \in R_M^*$. We say a state $s \in S$ is *reachable* from input (n, m) when there exists $n', m' \in \mathbb{N}$, s.t. $(\hat{s}, m, n) \to^* (s, n', m')$.

Finally, we can consider a machine as a partial function, we say that $M(m, n)$ returns (m', n') when $(\hat{s}, m, n) \to^* (\perp, n', m')$. Since complex data structure can be encoded as a pair of numbers using the classical Gödel numbers, we will abuse the notation to say $M(i)$ returns o for input i and output o of arbitrary type, not just pairs of numbers.

Given a two-counter machine with finite state set S, We define the set $\Sigma = S + \{\perp, a, b\}$ and the following commutable set $\ddot{\Sigma}$:

- the carrier is $\langle\Sigma| \cup |\Sigma\rangle$, where

$$\langle\Sigma| \triangleq \{\sigma_l \mid \sigma \in \Sigma\} \text{ and } |\Sigma\rangle \triangleq \{\sigma_r \mid \sigma \in \Sigma\};$$

- the commuting relation is $\langle\sigma| \sim |\sigma'\rangle$, where $\sigma, \sigma' \in \Sigma$.

We call primitives in $\langle\Sigma|$ *left primitives* and primitives in $|\Sigma\rangle$ *right primitives*. This definition of commutativity is similar to BiKA [AKL+22], however instead of using an underlying KA with two homomorphisms, we simply impose the commutativity onto the primitives. We consider Σ as a discrete commutable set, therefore we can define the free Kleene algebra $\mathsf{KA}(\Sigma)$ and the free KA with atomic commutative $\mathsf{KA}(\ddot{\Sigma})$.

There are three function we can define from Σ to $\mathsf{Wrd}(\ddot{\Sigma})$:

$$\langle -| : \Sigma \to \mathsf{Wrd}(\ddot{\Sigma}) \qquad |-\rangle : \Sigma \to \mathsf{Wrd}(\ddot{\Sigma}) \qquad \langle -\rangle : \Sigma \to \mathsf{Wrd}(\ddot{\Sigma})$$
$$\langle \sigma| \triangleq \sigma_l \qquad |\sigma\rangle \triangleq \sigma_r \qquad \langle \sigma\rangle \triangleq \sigma_l \cdot \sigma_r.$$

These maps can be lifted to monoidal homomorphism on words $\mathsf{Wrd}(\ddot{\Sigma}) \to \mathsf{Wrd}(\ddot{\Sigma})$: $\langle w|$ is the word w with all primitives replaced by corresponding left primitives and $|w\rangle$ is the word w with all primitives replaced by corresponding right primitives.

By composing $\langle -|, |-\rangle, \langle -\rangle$ with the natural monoidal embedding $\mathsf{Wrd}(\ddot{\Sigma}) \to \mathsf{KA}(\ddot{\Sigma})$, where the monoidal operation of $\mathsf{KA}(\ddot{\Sigma})$ is multiplication with identity 1, we can obtain functions in $\Sigma \to \mathsf{KA}(\ddot{\Sigma})$. Similarly, these maps can be lifted to KA homomorphisms $\mathsf{KA}(\Sigma) \to \mathsf{KA}(\ddot{\Sigma})$:

- $\langle e|$ and $|e\rangle$ will replace all the primitives in e with their respective left primitives or right primitives.
- $\langle e\rangle$ will produce two expression with *matching* left and right primitives.

We will abbreviate the multiplication $\langle e_1| \cdot |e_2\rangle$ as $\langle e_1|e_2\rangle$.

Example 1. *Consider $a \in \Sigma$, then $\langle a^*\rangle \in \mathsf{KA}(\ddot{\Sigma})$ and*

$$L(\langle a^*\rangle) = \{\langle a^n|a^n\rangle \mid n \in \mathbb{N}\}.$$

More generally, given an expression $e \in \mathsf{KA}(\Sigma)$, then $\langle e\rangle \in \mathsf{KA}(\ddot{\Sigma})$ and

$$L(\langle e\rangle) = \{\langle w|w\rangle \mid w \in L(e)\}.$$

For a word in $\mathsf{Wrd}(\ddot{\Sigma})$ we can always canonically separate it into its left and right components, this separation gives a normal form for all the words in $\mathsf{Wrd}(\ddot{\Sigma})$.

Definition 2. *For a word w, the* left component *$\langle w_l|$ is the word formed by all the left primitives in its original order, and the* right component *$|w_r\rangle$ is the word formed by all the right primitives in its original order. The concatenation of the left*

component and the right component is equal to the original word. Therefore, for all word $w, w' \in \mathsf{Wrd}(\ddot{\Sigma})$,

$$w = w' \iff w_r = w'_r \wedge w_l = w'_l.$$

Example 2. *Consider the following word*

$$w \triangleq \langle s|s'\rangle \cdot \langle a^m b^n | a^{m+1} b^n\rangle,$$

then its left component $\langle w_l| = \langle s a^m b^n|$ *and the right component is* $|w_r\rangle = |s' a^{m+1} b^n\rangle$. *The concatenation of right and left component is equal to the original word:*

$$\langle w_l | w_r\rangle = \langle s a^m b^n | s' a^{m+1} b^n\rangle = \langle s|s'\rangle \cdot \langle a^m b^n | a^{m+1} b^n\rangle.$$

We first show couple useful lemmas about $\mathsf{KA}(\Sigma)$ and $\mathsf{KA}(\ddot{\Sigma})$:

Lemma 4. *We first prove several lemmas that will be useful in our derivation. For all* $e, e_1, e_2, e'_1, e'_2 \in \mathsf{KA}(\Sigma)$

$$\langle e_1|e_2\rangle = |e_2\rangle \cdot \langle e_1|; \tag{2.1}$$

$$\langle e_1|e_2\rangle \leq \langle e'_1|e'_2\rangle \iff e_1 \leq e'_1 \wedge e_2 \leq e'_2; \tag{2.2}$$

$$\langle e^*\rangle \leq \langle e^*|e^*\rangle. \tag{2.3}$$

Proof. $\langle e_1|e_2\rangle = |e_2\rangle \cdot \langle e_1|$ can be derived by induction on structure of e_1, the only non-trivial case is the star case, which can be proven using the induction rule.

The equivalence

$$\langle e_1|e_2\rangle \leq \langle e'_1|e'_2\rangle \iff e_1 \leq e'_1 \wedge e_2 \leq e'_2$$

can be derived as follows: The $\Longleftarrow$ part can be shown by multiplication preserves order. The $\Longrightarrow$ part can be shown by looking at the language interpretation: if $L(e_1) \not\leq L(e'_1)$ or $L(e_2) \not\leq L(e'_2)$, then we can derive

$$L(\langle e_1|e_2\rangle) \not\leq L(\langle e'_1|e'_2\rangle)$$

$\langle e^* \rangle \leq \langle e^* | e^* \rangle$ can be shown by induction rule, because

$$\begin{aligned} &\langle e^* | e^* \rangle \geq 1, \\ &\langle e^* | e^* \rangle \geq \langle ee^* | ee^* \rangle = \langle e | e \rangle \cdot \langle e^* | e^* \rangle = \langle e \rangle \cdot \langle e^* | e^* \rangle, \end{aligned}$$

by induction rule $\langle e^* | e^* \rangle \geq \langle e^* \rangle$. □

We can encode components of the machine as Kleene Algebra terms.

Definition 3. *For simplicity, we will implicitly coerce all the configuration into an expression in* $\mathsf{KA}(\Sigma)$ *in the following way: for* $s \in S_\perp$*,*

$$(s, m, n) \mapsto s a^m b^n.$$

We interpret each instruction $i \in I_S$ *as an element* $[i] \in \mathsf{KA}(\ddot{\Sigma})$ *as follows.*

$$\begin{aligned} [\mathtt{Inc}(1, s)] &\triangleq |s\rangle\langle a^*\rangle|a\rangle\langle b^*\rangle \\ [\mathtt{Inc}(2, s)] &\triangleq |s\rangle\langle a^* b^*\rangle|b\rangle \\ [\mathtt{Dec}(1, s)] &\triangleq |s\rangle\langle a^*\rangle\langle a|\langle b^*\rangle + |s\rangle\langle b^*\rangle \\ [\mathtt{Dec}(2, s)] &\triangleq |s\rangle\langle a^* b^*\rangle\langle b| + |s\rangle\langle a^*\rangle \\ [\mathtt{If}(1, s_1, s_2)] &\triangleq |s_1\rangle\langle b^*\rangle + |s_2\rangle\langle a^+\rangle\langle b^*\rangle \\ [\mathtt{If}(2, s_1, s_2)] &\triangleq |s_1\rangle\langle a^*\rangle + |s_2\rangle\langle a^*\rangle\langle b\rangle^+ \\ [\mathtt{Halt}] &\triangleq |\perp\rangle\langle a^* b^*\rangle. \end{aligned}$$

The transition relation is encoded as $R_M \in \mathsf{KA}(\ddot{\Sigma})$*:*

$$R_M \triangleq \sum_{s \in S} \langle s| \cdot [\iota(s)].$$

The reason for such encoding is apparent when we look the language interpretation:

Corollary 7 (Language Soundness)**.** *Given a machine* $M \triangleq (S, \hat{s}, \iota)$*,* $c \in S \times \mathbb{N} \times \mathbb{N}$*, and* $c' \in S_\perp \times \mathbb{N} \times \mathbb{N}$*, the following equivalence holds*

$$[\![i]\!](m, n) = c' \iff \langle a^m b^n | c' \rangle \in L([i]); \tag{2.4}$$

$$c \to c' \iff \langle c | c' \rangle \in L(R_M). \tag{2.5}$$

Proof. The equivalence

$$[\![i]\!](m,n) = c' \iff \langle a^m b^n | c' \rangle \in L([i])$$

can be shown by looking at each case of instruction i, and explicitly compute the language model for each instruction.

Therefore,

$$\begin{aligned}
&(s,m,n) \to (s',m',n') \\
&\iff (s',m',n') = [\![\iota(s)]\!](m,n) \\
&\iff \langle a^m b^n | s' a^{m'} b^{n'} \rangle \in L([\iota(s)]) \\
&\iff \langle s a^m b^n | s' a^{m'} b^{n'} \rangle \in L(\langle s| \cdot [\iota(s)]) \\
&\iff \langle s a^m b^n | s' a^{m'} b^{n'} \rangle \in L(R_M).
\end{aligned}$$

□

Corollary 8. *Because the transition relation of any machine $M = (S, \hat{s}, \iota)$ is functional, therefore for all word $w \in L(Sa^*b^*)$, there exists a word $w' \in L(S_\perp a^* b^*)$ s.t. $\langle w | w' \rangle \in L(R_M)$.*

2.3.2 From Reachability to Undecidability

Our undecidability result relies on an equivalence between provability of a KA inequality and state reachability in a certain kind of machine. In order to obtain such equivalence, we will start with the provability of a single transition.

We will first define two useful terms. For a machine $M \triangleq (S, \hat{s}, \iota)$, and a subset of states $S' \subseteq S$, we can define all the configuration for S' including termination:

$$\begin{aligned}
&C_{S'} \in \mathsf{KA}(\Sigma) \\
&C_{S'} \triangleq S'_\perp a^* b^*.
\end{aligned}$$

And the term N_S representing left-right configuration mismatch:

$$
\begin{aligned}
&N_S \in \mathsf{KA}(\ddot{\Sigma}) \\
&N_S \triangleq \sum_{s \neq s' \in S} \langle s|s'\rangle(\langle a+b| + |a+b\rangle)^* && \text{state mismatch} \\
&\quad + \langle S\rangle\langle a^*\rangle(\langle a|^+ + |a\rangle^+)\langle b^*|b^*\rangle && \text{counter } a \text{ mismatch} \\
&\quad + \langle S\rangle\langle a^*b^*\rangle(\langle b|^+ + |b\rangle^+) && \text{counter } b \text{ mismatch}
\end{aligned}
$$

And we can use these terms to bound the encoding of transition R_M:

Lemma 5. *For all instructions $i \in I_S$ and transition $R_M \in \mathsf{KA}(\ddot{\Sigma})$, let S be the state set of M:*

$$[i] \leq \langle a^*b^*|C_S\rangle = \langle a^*b^*|S_\perp \cdot a^*b^*\rangle; \tag{2.6}$$

$$R_M \leq \langle C_S \mid C_S\rangle = \langle S_\perp \cdot a^*b^*|S_\perp \cdot a^*b^*\rangle. \tag{2.7}$$

Proof. The first inequality can be proven by looking at each case of i, and apply Inequality (2.3) when necessary.

The second inequality can be proven by unfolding the definition of R_M:

$$
\begin{aligned}
R_M &= \sum_{s\in S}\langle s| \cdot [\iota(s)] \\
&\leq \sum_{s\in S}\langle s| \cdot \langle a^*b^*|S_\perp \cdot a^*b^*\rangle && \text{by Inequality (2.6)} \\
&\leq \langle S \cdot a^*b^*|S_\perp \cdot a^*b^*\rangle \leq \langle C_S \mid C_S\rangle.
\end{aligned}
$$

□

With the above tools in place, we will establish the connection of provability and a single transition.

Theorem 10 (provability of single transition)**.** *In any machine M, $(s,m,n) \to (s',m',n')$ if and only if the following inequality is provable:*

$$|sa^mb^n\rangle R_M \leq \langle C_S\rangle \cdot |s'a^{m'}b^{n'}\rangle + N_S \cdot |C_S\rangle.$$

Proof. To prove the $\Longrightarrow$ direction, we will first unfold definition of the left-hand side:

$$\begin{aligned}
&|sa^m b^n\rangle R_M \\
&= |sa^m b^n\rangle \cdot (\sum_{s_1 \in S} \langle s_1| \cdot [\iota(s_1)]) \\
&= \sum_{s_1 \in S} |sa^m b^n\rangle \cdot \langle s_1| \cdot [\iota(s_1)] \\
&= \langle s|sa^m b^n\rangle \cdot [\iota(s)] + \sum_{s_1 \neq s} \langle s_1|sa^m b^n\rangle \cdot [\iota(s_1)].
\end{aligned}$$

It suffices to show the following inequalities:

$$\begin{aligned}
\langle s|sa^m b^n\rangle \cdot [\iota(s)] &\leq \langle C_S\rangle|s' a^{m'} b^{n'}\rangle + N_S|C_S\rangle; \\
\sum_{s_1 \neq s} \langle s_1|sa^m b^n\rangle \cdot [\iota(s_1)] &\leq \langle C_S\rangle|s' a^{m'} b^{n'}\rangle + N_S|C_S\rangle.
\end{aligned}$$

We show the second inequality first:

$$\begin{aligned}
&\sum_{s_1 \neq s} \langle s_1|sa^m b^n\rangle \cdot [\iota(s_1)] \\
&\leq \sum_{s_1 \neq s} \langle s_1|sa^m b^n\rangle \cdot \langle a^* b^*|C_S\rangle \\
&\leq \sum_{s \neq s' \in S} \langle s|s'\rangle(\langle a+b| + |a+b\rangle)^*|C_S\rangle \\
&\leq N_S|C_S\rangle \leq \langle C_S\rangle|s' a^{m'} b^{n'}\rangle + N_S|C_S\rangle,
\end{aligned}$$

To show the first inequality:

$$\langle s|sa^m b^n\rangle \cdot [\iota(s)] \leq \langle C_S\rangle|s' a^{m'} b^{n'}\rangle + N_S|C_S\rangle.$$

We need to look at all the cases for $\iota(s)$, we show the $\texttt{If}(1, s)$ case as examples:

- If $\iota(s) = \texttt{If}(1, s_1', s_2')$ and $m = 0$, then $(s, 0, n) \to (s_1', 0, n)$:

$$\begin{aligned}
&\langle s|sb^n\rangle \cdot [\iota(s)] \\
&= \langle s|sb^n\rangle \cdot (|s_1'\rangle\langle b^*\rangle + |s_2'\rangle\langle a^+\rangle\langle b^*\rangle) \\
&= \langle s|sb^n\rangle \cdot |s_1'\rangle \cdot (\sum_{i\le n}\langle b^i\rangle + \langle b^n\rangle \cdot \langle b^+\rangle) \\
&\quad + \langle s|sb^n\rangle \cdot |s_2'\rangle \cdot \langle a^+\rangle\langle b^*\rangle \\
&= (\sum_{i<n}\langle sb^i|sb^n\rangle|s_1'b^i\rangle) + \langle sb^n|sb^n\rangle \cdot |s_1'b^n\rangle \\
&\quad + \langle sb^nb|sb^n\rangle|s_1'b^nb\rangle\langle b^*\rangle \\
&\quad + \langle sa|sb^n\rangle \cdot |s_2'a\rangle \cdot \langle a^*b^*\rangle
\end{aligned}$$

  Notice:

$$\begin{aligned}
\sum_{i<n}\langle sb^i|sb^n\rangle|s_1'b^i\rangle &\le \langle S\rangle\langle a^*b^*\rangle|b^+\rangle \cdot |C_S\rangle \le N_S \cdot |C_S\rangle; \\
\langle sb^n|sb^n\rangle \cdot |s_1'b^n\rangle &\le \langle C_S\rangle \cdot |s_1'b^n\rangle; \\
\langle sb^nb|sb^n\rangle|s_1'b^nb\rangle\langle b^*\rangle &\le \langle sb^nb|sb^n\rangle|s_1'b^nb\rangle\langle b^*|b^*\rangle \\
&\le \langle sb^nbb^*|sb^n\rangle|s_1'b^nbb^*\rangle \\
&\le \langle S\rangle\langle a^*b^*\rangle\langle b^+| \cdot |C_S\rangle \le N_S \cdot |C_S\rangle; \\
\langle sa|sb^n\rangle \cdot |s_2'a\rangle \cdot \langle a^*b^*\rangle &\le \langle sa|sb^n\rangle \cdot |s_2'a\rangle \cdot \langle a^*|a^*\rangle\langle b^*|b^*\rangle \\
&\le \langle saa^*b^*|sb^n\rangle \cdot |s_2'ab^*a^*\rangle \\
&\le \langle S\rangle\langle a^*\rangle\langle a|^+\langle b^*|b^*\rangle \cdot |C_S\rangle \\
&\le N_S \cdot |C_S\rangle.
\end{aligned}$$

  Thus, we obtain the inequality we desire:

$$|sa^mb^n\rangle \cdot \langle s| \cdot [\iota(s)] \le \langle C_S\rangle \cdot |s_1'b^n\rangle + N_S|C_S\rangle.$$

- If $\iota(s) = \texttt{If}(1, s_1', s_2')$ and $m \neq 0$, then $(s, m, n) \to (s_2', m, n)$, and the proof is

similar to above:

$$\begin{aligned}
&\langle s|sa^m b^n\rangle \cdot [\iota(s)] \\
&= \langle s|sa^m b^n\rangle \cdot (|s_1'\rangle\langle b^*\rangle + |s_2'\rangle\langle a^+ b^*\rangle) \\
&= \langle s|sa^m b^n\rangle \cdot |s_1'\rangle\langle b^*\rangle \\
&\quad + \langle s|sa^m b^n\rangle \cdot |s_2'\rangle \cdot (\sum_{i\le m, j\le n} \langle a^i b^j\rangle + \langle a^m a^+ b^n b^+\rangle) \\
&= \langle s|sa^m b^n\rangle \cdot |s_1'\rangle\langle b^*\rangle \\
&\quad + (\sum_{i<m, j\le n} \langle sa^i b^j|sa^m b^n\rangle \cdot |s_2' a^i b^j\rangle) \\
&\quad + (\sum_{i=m, j\le n} \langle sa^i b^j|sa^m b^n\rangle \cdot |s_2' a^i b^j\rangle) \\
&\quad + \langle sa^m b^n|sa^m b^n\rangle \cdot |s_2' a^m b^n\rangle \\
&\quad + \langle sa^m a^+ b^n b^+|sa^m b^n\rangle \cdot |s_1' a^m a^+ b^n b^+\rangle
\end{aligned}$$

Then we obtain the following inequality

$$\begin{aligned}
\langle s|sa^m b^n\rangle \cdot |s_1'\rangle\langle b^*\rangle &\le \langle sb^*|sa^m b^n\rangle \cdot |s_1' b^*\rangle \\
&\le \langle S\rangle\langle a^*\rangle|a\rangle^+\langle b^*|b^*\rangle \cdot |C_S\rangle \\
&\le N_S|C_S\rangle; \\
(\sum_{i<m, j\le n} \langle sa^i b^j|sa^m b^n\rangle \cdot |s_2' a^i b^j\rangle) &\le \langle S\rangle\langle a^*\rangle|a\rangle^+\langle b^*|b^*\rangle \cdot |C_S\rangle \\
&\le N_S|C_S\rangle; \\
(\sum_{i=m, j\le n} \langle sa^i b^j|sa^m b^n\rangle \cdot |s_2' a^i b^j\rangle) &\le \langle S\rangle\langle a^* b^*\rangle|b\rangle^+ \cdot |C_S\rangle \\
&\le N_S|C_S\rangle; \\
\langle sa^m b^n|sa^m b^n\rangle \cdot |s_2' a^m b^n\rangle &\le \langle C_S\rangle \cdot |s_2' a^m b^n\rangle; \\
\langle sa^m a^+ b^n b^+|sa^m b^n\rangle \cdot |s_1' a^m a^+ b^n b^+\rangle &\le \langle S\rangle\langle a^*\rangle|a\rangle^+\langle b^*|b^*\rangle \cdot |C_S\rangle \\
&\le N_S|C_S\rangle
\end{aligned}$$

Thus, we obtain the inequality we desire:

$$|sa^m b^n\rangle \cdot \langle s| \cdot [\iota(s)] \le \langle C_S\rangle \cdot |s_2' a^m b^n\rangle + N_S|C_S\rangle.$$

The $\Longleftarrow$ direction can be shown by looking at the language model. If the inequality holds, then the language interpretation holds:

$$|sa^m b^n\rangle L(R_M) \subseteq L(\langle C_S\rangle) \cdot |s' a^{m'} b^{n'}\rangle + L(N_S) \cdot L(|C_S\rangle).$$

By Corollary 8, there exists a word $w \in L(R_M)$ s.t. its left component is $\langle sa^m b^n|$; we write the right component of w as $|w_r\rangle$. By Theorem 8, $w \leq R_M$; and by Inequalities (2.2) and (2.7),

$$\langle sa^m b^n|w_r\rangle \leq R_M \leq \langle C_S|C_S\rangle \Longrightarrow w_r \leq C_S.$$

Therefore, $w_r \in L(C_S)$ and by the definition of $L(N_S) \cdot L(|C_S\rangle)$,

$$|sa^m b^n\rangle \cdot \langle sa^m b^n|w_r\rangle = \langle sa^m b^n|sa^m b^n\rangle \cdot |w_r\rangle \notin L(N_S) \cdot L(|C_S\rangle).$$

Because of the language inclusion

$$|sa^m b^n\rangle L(R_M) \subseteq L(\langle C_S\rangle) \cdot |s' a^{m'} b^{n'}\rangle + L(N_S) \cdot L(|C_S\rangle),$$

we have

$$\langle sa^m b^n|sa^m b^n\rangle \cdot |w_r\rangle \in L(\langle C_S\rangle) \cdot |s' a^{m'} b^{n'}\rangle$$

therefore $w_r = s' a^{m'} b^{n'}$, and by Equivalence (2.5):

$$w = \langle sa^m b^n|s' a^{m'} b^{n'}\rangle \in R_M \Longrightarrow (s, m, n) \to (s', m', n').$$

□

We can further extend our previous result to establish a connection between provability and state reachability in certain type of machines. This result will be the core of our diagonal argument.

Theorem 11. *For a machine $M \triangleq (S, \hat{s}, \iota)$ and an input (m, n), if the set of reachable configurations from the input is finite, then a set $S' \subseteq S$ contains all the reachable states from input (m, n) if and only if the following inequality is provable:*

$$|\hat{s}a^n b^m\rangle R^*_M \leq \langle C^*_S\rangle|C_{S'}\rangle + \langle C^*_S\rangle N_S \langle C^*_S|C^*_S\rangle. \tag{2.8}$$

Proof. The $\Longrightarrow$ direction. We define the following term:

$$\mathsf{KA}(\Sigma) \ni C_r \triangleq \sum\{c_r \mid c_r \text{ is reachable}\}.$$

C_r is well-defined because there are only finitely many reachable configurations. By Theorem 10, assume $c_r \to c_r'$, we have the following inequality:

$$\begin{aligned} |c_r\rangle R_M &\leq \langle C_S|c_r'\rangle + N_S|C_S\rangle \\ &\leq \langle C_S|C_r\rangle + N_S|C_S\rangle \qquad\qquad c_r' \text{ is reachable} \end{aligned}$$

Since the above inequality is true for every reachable configuration c_r, then the following inequality is true:

$$|C_r\rangle R_M = \sum_{c_r} |c_r\rangle R_M \leq \langle C_S|C_r\rangle + N_S|C_S\rangle.$$

To show provability of Inequality (2.8), we will prove a stronger inequality:

$$|\hat{s}a^n b^m\rangle R_M^* \leq \langle C_S^*|C_r\rangle + \langle C_S^*\rangle N_S\langle C_S^*|C_S^*\rangle.$$

With Theorem 10 and inequality (2.7), we can derive the following two inequality:

$$\begin{aligned} \langle C_S^*|C_r\rangle R_M &\leq \langle C_S^*\rangle\langle C_S\rangle|C_r\rangle + \langle C_S^*\rangle N_S|C_S\rangle \\ &\leq \langle C_S^*\rangle|C_r\rangle + \langle C_S^*\rangle N_S\langle C_S^*|C_S^*\rangle \\ \langle C_S^*\rangle N_S\langle C_S^*|C_S^*\rangle R_M &\leq \langle C_S^*\rangle N_S\langle C_S^*|C_S^*\rangle\langle C_S|C_S\rangle \\ &\leq \langle C_S^*\rangle N_S\langle C_S^*|C_S^*\rangle \\ &\leq \langle C_S^*\rangle|C_r\rangle + \langle C_S^*\rangle N_S\langle C_S^*|C_S^*\rangle. \end{aligned}$$

Therefore,

$$\begin{aligned} &(\langle C_S^*\rangle|C_r\rangle + \langle C_S^*\rangle N_S\langle C_S^*|C_S^*\rangle)R_M \\ &\qquad \leq \langle C_S^*\rangle|C_r\rangle + \langle C_S^*\rangle N_S\langle C_S^*|C_S^*\rangle; \\ &|\hat{s}a^n b^m\rangle \leq |C_r\rangle \\ &\qquad \leq \langle C_S^*\rangle|C_r\rangle + \langle C_S^*\rangle N_S\langle C_S^*|C_S^*\rangle. \end{aligned}$$

By induction rule, we have the desired inequality:

$$\begin{aligned}|\hat{s}a^n b^m\rangle R_M^* &\leq \langle C_S^*\rangle |C_r\rangle + \langle C_S^*\rangle N_S \langle C_S^*|C_S^*\rangle \\ &\leq \langle C_S^*\rangle |C_{S'}\rangle + \langle C_S^*\rangle N_S \langle C_S^*|C_S^*\rangle.\end{aligned}$$

The $\Longleftarrow$ direction. We will first prove a small lemma: consider a configuration c_r that is reachable from input (m, n), we will show there exists a word $w \in L(\langle C_S^*\rangle)$ s.t.

$$w|c_r\rangle \in L(|\hat{s}a^n b^m\rangle)L(R_M)^*.$$

The theorem above is shown by induction on the number of steps to reach c_r:

- If c_r is reached in 0 steps, then

  $$c_r = (\hat{s}, m, n).$$

  In this case, $w = \epsilon$ and

  $$|c_r\rangle = |\hat{s}a^n b^m\rangle \in L(|\hat{s}a^n b^m\rangle)L(R_M)^*.$$

- If c_r is reached in $n+1$ steps, we find the configuration c_r' that is reached in n steps:

  $$(\hat{s}, m, n) \to^* c_r' \to c_r.$$

  By induction hypothesis, there is $w \in L(\langle C_S^*\rangle)$, s.t.

  $$w|c_r'\rangle \in L(|\hat{s}a^n b^m\rangle)L(R_M)^*.$$

  By Corollary 7,

  $$c_r' \to c_r \Longrightarrow \langle c_r'|c_r\rangle \in L(R_M).$$

  We have obtained our desired word:

  $$w|c_r'\rangle \cdot \langle c_r'|c_r\rangle = w\langle c_r'|c_r'\rangle |c_r\rangle \in L(|\hat{s}a^n b^m\rangle)L(R_M)^*.$$

Consider $w \in L(\langle C_S^*\rangle)$ s.t. $w|c_r\rangle \in L(|\hat{s}a^n b^m\rangle)L(R_M)^*$, by definition

$$w|c_r\rangle \notin L(\langle C_S^*\rangle N_S \langle C_S^*|C_S^*\rangle).$$

Because Inequality (2.8) holds, we have the following language inclusion:

$$L(|\hat{s}a^n b^m\rangle)L(R_M)^* \subseteq L(\langle C_S^*\rangle)L(|C_{S'}\rangle) + L(\langle C_S^*\rangle N_S \langle C_S^*|C_S^*\rangle).$$

Therefore,

$$w|c_r\rangle \in L(\langle C_S^*\rangle) \cdot L(|C_{S'}\rangle).$$

Finally, by unfolding the definition of $L(\langle C_S^*\rangle) \cdot L(|C_{S'}\rangle)$, we obtain $s' \in S$. □

Corollary 9. *For a machine $M \triangleq (S, \hat{s}, \iota)$ that always halt regardless of the input, it will have finitely many reachable states from any input. Therefore, for any input (m, n), S' contains all the reachable state from $(\hat{s}, m, n)$ if and only if inequality (2.8) is provable.*

We can then construct our diagonal argument. Assume that the equational theory of KA with atomic commutativity is decidable, there is a machine $P(S', M, M')$ that decides whether inequality (2.8) is provable when given machine $M \triangleq (S, \hat{s}, \iota)$ with the encoding of M' as input, and a subset of states $S' \subseteq S$.

Fix two distinct states s_1, s_2, we define the diagonal machine $D(M)$ as follows: let $M \triangleq (S, \hat{s}, \iota)$ and $S' \triangleq S \setminus \{s_1\}$,

- if $P(S', M, M)$ returns true, then we will go to state s_1 and returns true;
- if $P(S', M, M)$ returns false, then we will go to state s_2 and returns false.

Hence, state s_1 is reachable if and only if $P(S', M, M)$ returns true.

Then we employ the standard technique to feed the diagonal machine to itself. since we assumed equalities in KA with atomic commutativity is decidable, therefore D always terminates; hence we can enumerate all the possible the output of $D(D)$:

- If $D(D)$ returns true, then $P(S', D, D)$ returns true. By Corollary 9, $S' \triangleq S \setminus \{s_1\}$ contains all the reachable states of $D(D)$. However, by definition of D, s_1 is reachable when $P(S', D, D)$ is true, and $s_1 \notin S'$. Therefore, we obtain a contradiction.

- If $D(D)$ returns false, this means that $P(S', D, D)$ is false. By Corollary 9, $S' \triangleq S \setminus \{s_1\}$ do not contain all the reachable state. However, by definition of D, s_1 is not reachable in this case, and S' contains every state other than s_1. Hence, S' has to contain all the reachable state of $D(D)$. We got a contradiction again.

Therefore, our assumption that KA with atomic commutativity is decidable has to be false.

Corollary 10 (Incompleteness)**.** *There exists some commutable set X and two expression $e_1, e_2 \in \mathsf{KA}(X)$ s.t. $L(e_1) \subseteq L(e_2)$ but $e_1 \not\leq e_2$. In other words, there exists inequalities in the language interpretation that is not derivable using the theory.*

Proof. Assume that the language interpretation is complete, that is for all expression e_1, e_2

$$L(e_1) \subseteq L(e_2) \iff e_1 \leq e_2.$$

By definition of $\mathsf{KA}(X)$, $e_1 = e_2$ if and only if it can be proven using the theory of KA plus the commutativity in X, therefore deciding general equality is recursively-enumerable, by enumerating the proof.

However, since word inhabitance is decidable, language inclusion is co-recursively-enumerable, since we can simply check whether all words in $L(e_1)$ is in $L(e_2)$.

If the language inclusion is equivalent to inequalities in the theory, then the problem of inequalities in the theory is both recursively-enumerable and co-recursively-enumerable, hence decidable. This result contradicts our undecidability result for general inequality in KA with atomic commutativity hypotheses. Therefore, our assumption is false, and language interpretation is incomplete. □

2.4 Conclusion And Open Problem

In this paper we have shown that the word inhabitance problem in KA with commutativity hypotheses is decidable and complete, yet the general equalities are neither decidable nor complete. We believe this is the first known KA extension where the word inhabitance problem is decidable, yet the general equality is not.

Our method to show the decidability of word inhabitance problem involves using the matrix model to decompose a word into several components, which we believe is a novel technique in defining the empty word predicate and derivative in extensions of Kleene Algebra. This technique also yields straight-forward proof of soundness and the fundamental theorem.

However, there are still several important open problems: Several theorems leading to the undecidability result requires introspection on each case of the instructions, which leads to very long and tedious proof. We suspect some of these proofs, like the proof for Theorem 10, can be simplified by establishing more connection between the language interpretation and the free models. The exact complexity of KA with atomic commutativity is still unknown. In particular, we do not know whether the problem of deciding general equalities are RE-complete.

Chapter 3

Domain Reasoning In TopKAT

3.1 Completeness and Decidability of TopKAT

Our goal in this section is to construct a complete interpretation for TopKAT, thereby reducing its theory that of plain KAT. In other words, any equation between two TopKAT terms is logically equivalent to another equation between a pair of corresponding KAT terms. While this result is not new [ZdAG22a, ZdAG22b, PW22], we present a more streamlined proof that hinges on the universal properties of free KATs and TopKATs, without relying explicitly on language models. In any case, as a consequence of this result, we obtain the decidability of the equational theory of TopKAT as a corollary. Moreover, our technique helps us to construct complete models and interpretations simply by computation, as well as simplifying proofs of other results about TopKAT.

3.1.1 Reduction on free models

We first note that any free KAT over an alphabet K, B is also a TopKAT, where the largest element is $(\sum K)^*$. This fact can be seen by straightforward induction, see **??**.

Since every free KAT is a TopKAT, every KAT interpretation $I : \mathsf{KAT} \to \mathcal{M}$ induces a sub-KAT $\mathbf{Im}(I) \subseteq \mathcal{M}$, and this sub-KAT happens to be a *TopKAT*. Indeed, the image of $(\sum K)^*$ in $\mathcal{M}$ is the largest element of $\mathbf{Im}(I)$, and the restriction $I : \mathsf{KAT} \to \mathbf{Im}(I)$ is a homomorphism of TopKATs.

This gives us a powerful tool to construct complete TopKAT interpretations. Since we already know that the KAT interpretations $G : \mathsf{KAT} \to \mathcal{G}$ and $h \circ G : \mathsf{KAT} \to \mathbf{Im}(h)$ are TopKAT homomorphisms and are injective, we can construct complete TopKAT interpretations by *composition*, simply by constructing an injective TopKAT interpretation r of type $\mathsf{TopKAT}(K, B) \to \mathsf{KAT}(K_\top, B)$:

$$\mathsf{TopKAT}(K,B) \xrightarrow{r} \mathsf{KAT}(K_\top, B) \xrightarrow{G} \mathcal{G}_{K_\top,B}, \mathsf{TopKAT}(K,B) \xrightarrow{r} \mathsf{KAT}(K_\top, B) \xrightarrow{G} \mathcal{G}_{K_\top,B} \xrightarrow{h} \mathbf{Im}(h).$$

Since an interpretation is determined by its action on primitives, we can define r just by specifying its behavior on $K + B$:

$$r : K + B \to \mathsf{KAT}(K_\top, B)$$
$$r(p) \triangleq p \qquad\qquad p \in K + B.$$

Since this is the only such r with this property, we can check that the homomorphism r coincides with the *reduction maps* of the same name in previous works [ZdAG22a, PW22]. More concretely, we can picture r as simply replacing the symbol $\top$ in a TopKAT term with $(\sum K_\top)^*$, the largest element in $\mathsf{KAT}(K_\top, B)$.

We will show that r is injective by constructing a left inverse for it. In fact, the left inverse is a very principled map.

Lemma 6. *The natural map $[-]_\top : \mathsf{KAT}(K_\top, B) \to \mathsf{TopKAT}(K, B)$, where each term is mapped to its corresponding equivalence class, is a TopKAT homomorphism.*

Proof. First, note that this map is well-defined. Indeed, a raw KAT term over $K_\top, B$ can be viewed as a raw TopKAT term over K, B. Moreover, since the theory of TopKAT extends that of KAT with an equation, any two equivalent KAT terms are also equivalent when seen as KAT terms. Thus, the identity on raw terms can be lifted to equivalence classes.

Since each term in $\mathsf{KAT}(K_\top, B)$ is mapped to its identical term in $\mathsf{TopKAT}(K, B)$, then it is clear that is preserves all the operations except perhaps for $\top$.

All we need to show is that $[-]_\top$ preserves the top element, that is $[(\sum K_\top)^*]_\top$ is

the largest element in $\mathsf{TopKAT}(K, B)$, which suffices to prove $(\sum K_\top)^* \geq \top$.

$$(\sum K_\top)^* \geq \sum K_\top = \top + \sum K \geq \top.$$

□

Lemma 7 (reduction)**.** $[-]_\top$ *is the right inverse of* r*:* $[-]_\top \circ r = id_{\mathsf{TopKAT}(K,B)}$*. More explicitly for all* $t \in \mathsf{TopKAT}(K, B)$*:*

$$\mathsf{TopKAT}(K, B) \vDash [r(t)]_\top = t.$$

Proof. Since $[-]_\top \circ r : \mathsf{TopKAT}(K, B) \to \mathsf{TopKAT}(K, B)$ is a TopKAT interpretation, hence the action on the primitive uniquely determines the interpretation: because both r and $[-]_\top$ do not change the primitives, therefore $[-]_\top \circ r$ is the identity interpretation on $\mathsf{TopKAT}(K, B)$. □

The above lemma matches one of the soundness condition of reductions in previous works [ZdAG22a, KS97, PRW21], which was typically proven by induction on the structure of terms. The induction approach involves case analysis on all operations of TopKAT [ZdAG22a]; whereas our approach determines the equality simply by computing the action of $[-]_\top \circ r$ on primitives.

Since r has a right inverse, it is a complete TopKAT interpretation:

$$\mathsf{TopKAT}(K, B) \vDash t_1 = t_2 \iff r(t_1) = r(t_2),$$

With the completeness of r, we can already show the complexity of TopKAT.

Corollary 11 (Complexity)**.** *Given two terms* $t_1, t_2 \in \mathsf{TopKAT}(K, B)$*, deciding whether these two terms are equal is PSPACE-complete.*

Proof. Since deciding KAT equality is a sub-problem of deciding TopKAT equality, and KAT equality is PSPACE-hard [CKS99], Thus TopKAT equality is PSPACE-hard.

To decide the equality of t_1, t_2, we first remove all the redundant primitive that does not appear in t_1, t_2 from the alphabet K, B. Then we compute $r(t_1)$ and $r(t_2)$,

each taking polynomial space (of $|t_1| + |t_2|$) to store; and we use the standard algorithm [CKS99] to decide whether $r(t_1) = r(t_2)$ in $\mathsf{KAT}(K_\top, B)$, this will also take polynomial space. Hence, the decision procedure for TopKAT equality in PSPACE.

Thus deciding TopKAT equality is PSPACE-complete. □

3.1.2 Complete Model For Free

Designing complete interpretations and models was not always easy. In fact, in previous works [ZdAG22b], the authors made a mistake in the definition of language TopKAT, which was fixed later [ZdAG22a] by suggestion of Pous et al. [PW22]. However, with the results in Section 3.1.1, we can construct the complete interpretation just by composition, and compute the complete model by computing the range of the complete interpretation.

We already know that there are two complete interpretations of TopKAT defined as follows:

$$\mathsf{TopKAT}(K, B) \xrightarrow{r} \mathsf{KAT}(K_\top, B) \xrightarrow{G} \mathcal{G}_{K_\top, B}, \mathsf{TopKAT}(K, B) \xrightarrow{r} \mathsf{KAT}(K_\top, B) \xrightarrow{G} \mathcal{G}_{K_\top, B} \xrightarrow{h} \mathbf{Im}(h),$$

with a complete language model $\mathcal{G}_{K_\top, B}$, and a complete model consists of relations $\mathbf{Im}(h)$.

The operations in these models can be recovered by computing these maps. For example, the multiplication operation in the language TopKAT can be computed as follows:

$$G \circ r(t_1 \cdot t_2) = G(r(t_1) \cdot r(t_2)) = G(r(t_1)) \diamond G(r(t_2)).$$

Since r do not change the multiplication operation, the multiplication in the language TopKAT is the same as in language KAT. In fact, as r do not change any operation in KAT, most operations in language TopKAT is the same as language KAT. Thus, we only need to figure out the top element in language TopKAT.

The top element in language TopKAT can be computed in the same fashion:

$$G \circ r(\top) = G((\sum K_\top)^*) = GS_{K_\top, B},$$

i.e. the top element is just the complete language.

Corollary 12. *The language TopKAT inherits all the operations in language KAT, except the top element is defined as the full language. And such model is complete with $G \circ r$ as a complete interpretation.*

In the same way, we know that complete model consisting of relations (a.k.a. general relational TopKAT) will have the same operations as relational KATs. However, in this case the characterization the computed top: $h \circ G \circ r(\top)$ is not as simple as the full language; but we know it is the largest relation in the range of $h \circ G \circ r$:

Corollary 13. *The general relational TopKAT inherits all the operations in relational KAT, except the top element is the largest relation. And such model is complete with $h \circ G \circ r$ as a complete interpretation.*

Finally, to investigate whether we can use general relational TopKAT to encode incorrectness logic, we will provide a short proof that general relational TopKATs are as expressive as relational KATs [ZdAG22a]; that is, every property on relations that can be encoded using general relational TopKAT, is already encodable in the relational KAT. Hence, adding a top element do not give extra expressive power in general relational TopKAT.

The original proof [ZdAG22a, Lemma 2] encodes every TopKAT term using KAT term, and then use two pages to prove the soundness of such encoding. Here we show such encoding is simply the reduction r.

Corollary 14 (Expressiveness of general relational TopKAT)**.** *Given an alphabet K, B, an n-ary predicate P on relations, the predicate P over primitives $p_1, p_2, ..., p_n \in K$ is expressible in general relational TopKAT if and only if it is expressible in relational KAT.*

Formally, it suffices to show that if there exist two terms $t_1, t_2 \in \mathsf{TopKAT}(K, B)$ *s.t. for any general relational interpretation* $I_\top$*:*

$$I_\top(t_1) = I_\top(t_2) \iff P(I_\top(p_1), I_\top(p_2), ..., I_\top(p_n));$$

then take an arbitrary relational KAT interpretation I*:*

$$I(r(t_1)) = I(r(t_2)) \iff P(I(p_1), I(p_2), ..., I(p_n)).$$

Proof. Take an arbitrary relational KAT interpretation I from $\mathsf{KAT}(K_\top, B)$. Notice $\mathbf{Im}(I)$, the range of I, is a relational KAT with a largest element (I of the largest element in $\mathsf{KAT}(K_\top, B)$); hence $\mathbf{Im}(I)$ is a general relational TopKAT. Then I is a TopKAT homomorphism from $\mathsf{KAT}(K_\top, B)$ to its range: I preserves all the operation of KAT, since it is a KAT interpretation; and preserves the top element, since homomorphism preserves order.

Then we can construct $I \circ r : \mathsf{TopKAT}(K, B) \to \mathbf{Im}(I)$, a general relational interpretation. And since r does not modify primitives, therefore

$$\forall p_i \in K, I \circ r(p_i) = I(r(p_i)) = I(p_i).$$

Finally,

$$\begin{aligned} I(r(t_1)) = I(r(t_2)) &\iff I \circ r(t_1) = I \circ r(t_2) \\ &\iff P(I \circ r(p_1), ..., I \circ r(p_n)) \quad I \circ r \text{ is a TopGREL interpretation} \\ &\iff P(I(p_1), ..., I(p_n)) \qquad\qquad \forall p_i \in K, I \circ r(p_i) = I(p_i) \end{aligned}$$

□

Since the image of I is not necessary a relational TopKAT, where the top element is interpreted as the complete relation, The above trick do not work for relational TopKAT. It is also known that relational TopKAT is strictly more expressive than general relational TopKAT, since relational TopKAT can encode incorrectness logic, where general relational TopKAT cannot [ZdAG22a].

3.2 Domain Completeness

In general, TopKAT is not complete over relational models, which are crucial for program-analysis applications [ZdAG22a]. However, it was later showed that we *can* obtain a complete theory for relational models by simply adding the axiom $p\top p \geq p$ to the theory of TopKAT [PW22]. We aim to investigate how much these two theories differ in expressive power for program reasoning. In particular, the encoding of incorrectness logic in TopKAT [ZdAG22a] relies only on the ability of TopKAT to compare the domain and codomain of two relations. This raises the question of whether TopKAT suffices for proving such properties; that is, whether the following completeness results hold: for $t_1, t_2 \in \mathsf{KAT}(K, B)$ (i.e. $\top$ does not appear in t_1 and t_2)

$$\mathrm{REL} \models \mathrm{cod}(t_1) \geq \mathrm{cod}(t_2) \iff \mathsf{TopKAT} \models \top t_1 \geq \top t_2 \quad \text{codomain completeness}$$

$$\mathrm{REL} \models \mathrm{dom}(t_1) \geq \mathrm{dom}(t_2) \iff \mathsf{TopKAT} \models t_1 \top \geq t_2 \top \quad \text{domain complete}$$

In this section, we prove that these equivalences hold, even without the additional axiom $p\top p \geq p$. However, they do *not* hold if we allow terms that contain top: for example, let $t_1 \triangleq p\top p$, and $t_2 \triangleq p$. Since $p\top p \geq p$ holds in relational TopKAT, thus $\mathrm{dom}(p\top p) \geq \mathrm{dom}(p)$; but $p\top p\top \geq p\top$ is not provable in TopKAT, since the inequality is not true in the language interpretation: $G \circ r(p\top p\top)$ gives the language which contains at least two p and starts with action p, which does not contain $G \circ r(p\top)$, where only starting with action p is required. The incompleteness of codomain comparison can also be shown using the same example.

The core idea to prove codomain completeness for KAT terms is to construct a specific relational interpretation, where its codomain is equivalent to the complete

TopKAT interpretation $G \circ r$:

$$\mathrm{cod}(h \circ i \circ G(t)) = G \circ r(\top t),$$

where i is the natural inclusion homomorphism $i : \mathcal{G}_{K,B} \hookrightarrow \mathcal{G}_{K_\top,B}$, that maps every language to itself; and h is the classical embedding of language KAT into relational KAT [KS97]. Although i will not change the outcome of G, it will add a new primitive action $\top$ to the alphabet, hence changing the outcome of h. Such addition will equate the codomain of $h \circ i \circ G(t)$ with the complete TopKAT interpretation $G \circ r$ of $\top t$. The proof of this equality is by simply computing both sides of the equation, and is in the appendix as **??**;

With the above equality obtained, the codomain completeness can be shown as follows.

Theorem 12 (Codomain completeness)**.** *Given two terms $t_1, t_2 \in \mathsf{KAT}(K, B)$ (i.e. term without $\top$), then codomain comparison is complete:*

$$\mathrm{REL} \vDash \mathrm{cod}(t_1) \geq \mathrm{cod}(t_2) \iff \mathsf{TopKAT} \vDash \top t_1 \geq \top t_2.$$

Proof. Given the natural inclusion homomorphism: $i : \mathsf{KAT}(K, B) \to \mathsf{KAT}(K_\top, B)$, we show that the following are equivalent:

1. $\mathrm{REL} \vDash \mathrm{cod}(t_1) \geq \mathrm{cod}(t_2)$.
2. $\mathrm{cod}(h \circ i \circ G(t_1)) \geq \mathrm{cod}(h \circ i \circ G(t_2))$.
3. $\mathsf{TopKAT} \vDash \top t_1 \geq \top t_2$.

We first show that $1 \implies 2$, since $\mathrm{REL} \vDash \mathrm{cod}(t_1) \geq \mathrm{cod}(t_2)$, means $\mathrm{cod}(I(t_1)) \geq \mathrm{cod}(I(t_2))$ for all relational KAT interpretation I, and $h \circ i \circ G$ is a relational KAT interpretation, so this is true.

We show $2 \Longrightarrow 3$, which uses the equality discussed above, and proved in **??**:

$$\begin{aligned}
& \mathrm{cod}(h \circ i \circ G(t_1)) \geq \mathrm{cod}(h \circ i \circ G(t_2)) && \\
\iff & G \circ r(\top t_1) \geq G \circ r(\top t_2) && \textbf{??} \\
\iff & \mathsf{TopKAT} \vDash \top t_1 \geq \top t_2. && \text{Completeness of } G \circ r
\end{aligned}$$

Finally, we show $3 \Longrightarrow 1$, since $\mathsf{TopKAT} \vDash \top t_1 \geq \top t_2 \Longrightarrow \mathrm{TopREL} \vDash \top t_1 \geq \top t_2$, and because $\mathrm{TopREL} \vDash \top t_1 \geq \top t_2 \Longrightarrow \mathrm{REL} \vDash \mathrm{cod}(t_1) \geq \mathrm{cod}(t_2)$ (**??**), therefore $\mathsf{TopKAT} \vDash \top t_1 \geq \top t_2 \Longrightarrow \mathrm{REL} \vDash \mathrm{cod}(t_1) \geq \mathrm{cod}(t_2)$. □

The domain completeness can be obtained from exploring the properties of relational converse. Recall the definition of converse operator on relation $R^\vee \triangleq \{(b, a) \mid (a, b) \in R\}$, Then by unfolding the definition, we have $\mathrm{cod}(R) = \mathrm{dom}(R^\vee)$.

We can extend this converse operations to relational KAT interpretation. Given a interpretation $I : \mathsf{KAT}(K, B) \to X \times X$, we can define $I^\vee$ by conversing the action of I on primitives:

$$I^\vee : K + B \to X \times X$$
$$I^\vee(p) \triangleq I(p)^\vee.$$

The converse operator $(-)^\vee : (\mathsf{KAT}(K, B) \to X \times X) \to (\mathsf{KAT}(K, B) \to X \times X)$ is an isomorphism on relational interpretation, where it is own inverse. This can be verified by looking at the action of $(-)^\vee \circ (-)^\vee(I)$ on the primitives:

$$\forall p \in K + B, (I^\vee)^\vee(p) = ((I(p))^\vee)^\vee = I(p) \Longrightarrow (-)^\vee \circ (-)^\vee = id_{(\mathsf{KAT}(K,B) \to X \times X)}.$$

Therefore, for an arbitrary relational KAT interpretation I, we can find a relational KAT interpretation I' s.t. $I'^\vee = I$; then if $\mathrm{dom}(I'^\vee(t_1)) = \mathrm{dom}(I'^\vee(t_2))$, we can derive

$$\mathrm{cod}(I(t_1)) = \mathrm{dom}(I'^\vee(t_1)) = \mathrm{dom}(I'^\vee(t_2)) = \mathrm{cod}(I(t_2)).$$

Let I' and I ranges over all relational KAT interpretations, by the previous result, we have

$$\forall I', \mathrm{dom}(I'(t_1)) = \mathrm{dom}(I'(t_2)) \Longrightarrow \forall I, \mathrm{cod}(I(t_1)) = \mathrm{cod}(I(t_2)) \Longrightarrow t_1\top = t_2\top.$$

Thus, we obtained the non-trivial side of domain completeness result, and the other side is a direct consequence of **??**.

Theorem 13 (Domain Completeness)**.** *Given two terms $t_1, t_2 \in \mathsf{KAT}(K, B)$ (i.e. term without $\top$), then domain comparison is complete:*

$$\mathrm{REL} \vDash \mathrm{cod}(t_1) \geq \mathrm{cod}(t_2) \Longleftrightarrow \mathsf{TopKAT} \vDash \top t_1 \geq \top t_2.$$

3.3 A Coalgebraic Theory of TopKAT

Coalgebraic decision procedure has shown great promise in real-world applications [FKM+15, SKK+19, Pou15]. In this section we will develop the coalgebraic theory of TopKAT; it should come as no surprise that such coalgebraic theory can be obtained trivially using the reduction result, and produce a efficient decision procedure. Further demonstrating the importance of such reduction result.

Later we will show that the coalgebraic decision procedure for determining reachability properties is as efficient as breath-first search. This demonstrate that our procedure is general enough to cover all equation of TopKAT; yet when specialized to domain comparison, it is as efficient as specialized algorithms.

3.3.1 Kleene Coalgebra with Tests

A Kleene Coalgebra with tests (KCT) $\mathcal{K}$ over a alphabet (K, B) consists of two family of operations indexed by $\alpha \in 1_{\mathcal{G}}$ and $\alpha p \in 1_{\mathcal{G}} \times K$:

$$\epsilon_\alpha : \mathcal{K} \to 2, \delta_{\alpha p} : 1_{\mathcal{G}} \times K \to \mathcal{K} \to \mathcal{K},$$

where $2 \triangleq \{0, 1\}$ is the two-element set. By the natural isomorphism $(\mathcal{K} \to 2) \times (\mathcal{K} \to \mathcal{K}) \cong \mathcal{K} \to 2 \times \mathcal{K}$, these two operations can be combined into one, denoted as $\langle \epsilon_\alpha, \delta_{\alpha p} \rangle : \mathcal{K} \to 2 \times \mathcal{K}$. These operations are inspired by the structure of guarded languages, intuitively, ϵ_α computes whether α is contained with in a guarded language, and $\delta_{\alpha p}$ computes the remaining language after αp has been removed from the front.

This intuition can be made concrete by the finality of the guarded language model. Guarded language model $\mathcal{G}_{K,B}$ forms a KCT when equipped with the following operations:

$$\epsilon_\alpha : \mathcal{G} \to 2 \qquad\qquad \delta_{\alpha p} : \mathcal{G} \to \mathcal{G}$$
$$\epsilon_\alpha(S) \triangleq (\alpha \in S), \qquad\qquad \delta_{\alpha p}(S) \triangleq \{s \mid \alpha p s \in S\}.$$

Notice that these two operation exactly corresponds to our intuitions. Furthermore, the guarded language over K, B is final in all KCT over K, B: for all KCT $\mathcal{K}$, there exists a unique homomorphism $h : \mathcal{K} \to \mathcal{G}$:

$$\begin{array}{ccc} \mathcal{K} & \xrightarrow{\quad h \quad} & \mathcal{G} \\ {\scriptstyle \langle \epsilon_\alpha, \delta_{\alpha p} \rangle}\downarrow & & \downarrow{\scriptstyle \langle \epsilon_\alpha, \delta_{\alpha p} \rangle} \\ 2 \times \mathcal{K} & \xrightarrow[(id,h)]{} & 2 \times \mathcal{G} \end{array}$$

This suggests that for all KCT $\mathcal{K}$, we can correspond each of its element to a language, and the commutativity of the diagram implies that the $\epsilon_\alpha, \delta_{\alpha p}$ in $\mathcal{K}$ will perform exactly the operation $\epsilon_\alpha, \delta_{\alpha p}$ on their corresponding languages.

In particular, the free $\mathsf{KAT}(K, B)$ with the operations E_α and $D_{\alpha p}$ defined by Kozen [Koz08, Section 4.2] from a KCT, and the unique coalgebra homomorphism

from the $\mathsf{KAT}(K, B)$ to $\mathcal{G}_{K,B}$ is exactly the language interpretation:

$$\begin{array}{ccc} \mathsf{KAT}(K,B) & \xrightarrow{G} & \mathcal{G}_{K,B} \\ {\scriptstyle \langle E_\alpha, D_{\alpha p}\rangle}\downarrow & & \downarrow{\scriptstyle \langle \epsilon_\alpha, \delta_{\alpha p}\rangle} \\ 2 \times \mathsf{KAT}(K,B) & \xrightarrow[(id,G)]{} & 2 \times \mathcal{G}_{K,B} \end{array} \tag{3.1}$$

By finality of the language interpretation, two KAT terms are bisimilar if and only if they have the same guarded language interpretations [Sil10, Theorem 2.2.6, Theorem 2.2.7]. And by completeness of the guarded language model [KS97], two KAT terms are bisimilar if and only if they are equal in the free KAT. Finally, Kozen showed that bisimulation on the coalgebra $\mathsf{KAT}(K, B)$ gives a PSPACE algorithm for deciding KAT equalities [Koz08, Section 6].

3.3.2 Coalgebra From Reduction

Similar to the case of KCT, our definition of TopKAT coalgebra is based on the structure of guarded language model for TopKATs. Although the guarded language model for TopKATs are also sets of guarded strings, the $\top$ symbol is always in the action alphabet, which differentiates the structure of guarded languages for KATs and TopKATs. To handle the $\top$ symbol, we need an additional family of operations $\delta_{\alpha\top} : \mathcal{K} \to \mathcal{K}$ to consume the $\top$ symbol.

Definition 4 (TopKAT coalgebra)**.** *A TopKAT coalgebra $\mathcal{K}$ over K, B consists of three family of operations, indexed by $\alpha \in 1_\mathcal{G}$, $\alpha p \in 1_\mathcal{G} \times K$ and $\alpha \in 1_\mathcal{G}$ respectively:*

$$\epsilon_\alpha : \mathcal{K} \to 2, \delta_{\alpha p} : \mathcal{K} \to \mathcal{K}, \delta_{\alpha\top} : \mathcal{K} \to \mathcal{K}.$$

Where $\delta_{\alpha\top}$ is an operation that consumes an atom α and a $\top$ symbol.

Despite the slight differences in the definition, TopKAT coalgebras have strong connections to Kleene coalgebras with tests:

Corollary 15. *Because of the natural isomorphism $(1_\mathcal{G} \times K \to X) \times (1_\mathcal{G} \to X) \cong 1_\mathcal{G} \times (K+1) \to X \cong 1_\mathcal{G} \times K_\top \to X$, three families of operations in TopKAT coalgebra*

can be combined into two families indexed by $\alpha \in 1_{\mathcal{G}}$ *and* $\alpha p' \in 1_{\mathcal{G}} \times (K_\top)$*:*

$$\epsilon_\alpha : \mathcal{K} \to 2, \delta_{\alpha p'} : \mathcal{K} \to \mathcal{K}.$$

Therefore TopKAT coalgebras over K, B *are exactly the KCTs over* $K_\top, B$*.*

Since $\mathcal{G}_{K_\top,B}$ *is the final KCT over* $K_\top, B$*, it is the final TopKAT coalgebra over* K, B*.*

The two families coalgebra operations on the free TopKAT is defined as follow, where the first operation is indexed by $\alpha \in 1_{\mathcal{G}}$ and the second by $\alpha p \in 1_{\mathcal{G}} \times K_\top$:

$$E_\alpha \circ r : \mathsf{TopKAT}(K,B) \xrightarrow{r} \mathsf{KAT}(K_\top, B) \xrightarrow{E_\alpha} 2,$$
$$[-]_\top \circ D_{\alpha p} \circ r : \mathsf{TopKAT}(K,B) \xrightarrow{r} \mathsf{KAT}(K_\top, B) \xrightarrow{D_{\alpha p}} \mathsf{KAT}(K_\top, B) \xrightarrow{[-]_\top} \mathsf{TopKAT}(K,B).$$

The connection to language interpretation of TopKAT is apparent:

Theorem 14. *The language interpretation of TopKAT is the unique coalgebra homomorphism from* $\mathsf{TopKAT}(K,B)$ *to* $\mathcal{G}_{K_\top,B}$*:*

$$\begin{array}{ccc} \mathsf{TopKAT}(K,B) & \xrightarrow{G \circ r} & \mathcal{G}_{K_\top,B} \\ {\scriptstyle \langle E_\alpha \circ r, [-]_\top \circ D_{\alpha p} \circ r \rangle}\downarrow & & \downarrow{\scriptstyle \langle \epsilon_\alpha, \delta_{\alpha p} \rangle} \\ 2 \times \mathsf{TopKAT}(K,B) & \xrightarrow[(id, G \circ r)]{} & 2 \times \mathcal{G}_{K_\top,B} \end{array}$$

Proof. The uniqueness is trivial by the finality of $\mathcal{G}_{K_\top,B}$. We only need to show commutativity of the diagram, i.e. $(id, G \circ r) \circ \langle E_\alpha \circ r, D_{\alpha p} \circ r \rangle = \langle \epsilon_\alpha, \delta_{\alpha p} \rangle \circ G \circ r$, which is equivalent to show

$$E_\alpha \circ r = \epsilon_\alpha \circ G \circ r \text{ and } G \circ r \circ [-]_\top \circ D_{\alpha p} \circ r = \delta_{\alpha p} \circ G \circ r.$$

First by eq. (3.1), $E_\alpha = \epsilon_\alpha \circ G \implies E_\alpha \circ r = \epsilon_\alpha \circ G \circ r$.

Then because $r \circ [-]_\top = id_{\mathsf{TopKAT}(K,B)}$ and eq. (3.1), $G \circ r \circ [-]_\top \circ D_{\alpha p} \circ r = G \circ D_{\alpha p} \circ r = \delta_{\alpha p} \circ G \circ r$. □

Similar to the case in KAT, by finality of the guarded language interpretation of TopKAT two TopKAT terms are bisimilar if and only if they have the same guarded

language interpretations [Sil10, Theorem 2.2.6, Theorem 2.2.7]. And by completeness of the guarded language model [KS97], two KAT terms are bisimilar if and only if they are equal in the free KAT.

Finally, since every TopKAT coalgebra is over K, B is a KCT over $K_\top, B$, and because there is a PSPACE bisimulation algorithm for KCT [Koz08, Section 6], TopKAT bisimulation can also be computed in PSPACE.

We summarize the above two results in the following theorem:

Theorem 15 (Completeness and Decidability)**.** *Two TopKAT terms are equal if and only if they are bisimilar, and the bisimilarity of two TopKAT term can be decided in PSPACE.*

Chapter 4

Decompilation Verification With GKAT

4.1

Appendix A

Proof of xyz

This is the appendix.

References

Timos Antonopoulos, Eric Koskinen, Ton Chanh Le, Ramana Nagasamudram, David A. Naumann, and Minh Ngo. An algebra of alignment for relational verification. (arXiv:2202.04278), Jul 2022. arXiv:2202.04278 [cs].

Ernie Cohen, Dexter Kozen, and Frederick Smith. The complexity of kleene algebra with tests. Jul 1999.

Amina Doumane, Denis Kuperberg, Damien Pous, and Pierre Pradic. Kleene algebra with hypotheses. In *22nd International Conference on Foundations of Software Science and Computation Structures (FoSSaCS)*, Proc. FoSSaCS 2019, Prague, Czech Republic, 2019. Springer.

Nate Foster, Dexter Kozen, Mae Milano, Alexandra Silva, and Laure Thompson. A coalgebraic decision procedure for netkat. In *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '15, page 343–355, New York, NY, USA, Jan 2015. Association for Computing Machinery.

Dexter Kozen and Konstantinos Mamouras. Kleene algebra with equations. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming*, Lecture Notes in Computer Science, page 280–292, Berlin, Heidelberg, 2014. Springer.

D. Kozen. A completeness theorem for kleene algebras and the algebra of regular events. *Information and Computation*, 110(2):366–390, May 1994.

Dexter Kozen. *Myhill-Nerode Relations on Automatic Systems and the Completeness of Kleene Algebra*, volume 2010 of *Lecture Notes in Computer Science*, page 27–38. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

Dexter Kozen. On the complexity of reasoning in kleene algebra. *Information and Computation*, 179(2):152–162, Dec 2002.

Dexter Kozen. On the coalgebraic theory of kleene algebra with tests. March 2008. Accepted: 2008-03-14T18:49:38Z.

Daniel Krob. Complete systems ofb-rational identities. *Theoretical Computer Science*, 89(2):207–343, Oct 1991.

Dexter Kozen and Frederick Smith. *Kleene algebra with tests: Completeness and decidability*, volume 1258 of *Lecture Notes in Computer Science*, page 244–259. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

Dexter Kozen and Alexandra Silva. Left-handed completeness. *Theoretical Computer Science*, 807:220–233, Feb 2020.

Joachim Lambek. How to program an infinite abacus. *Canadian Mathematical Bulletin*, 4(3):295–302, September 1961.

Marvin L. Minsky. Recursive unsolvability of post's problem of "tag" and other topics in theory of turing machines. *The Annals of Mathematics*, 74(3):437, November 1961.

Marvin L. Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Inc., USA, 1967.

Damien Pous. Symbolic algorithms for language equivalence and kleene algebra with tests. In *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '15, page 357–368, New York, NY, USA, January 2015. Association for Computing Machinery.

Damien Pous, Jurriaan Rot, and Jana Wagemaker. On tools for completeness of kleene algebra with hypotheses. In *Relational and Algebraic Methods in Computer Science: 19th International Conference, RAMiCS 2021, Marseille, France, November 2–5, 2021, Proceedings*, page 378–395, Berlin, Heidelberg, Nov 2021. Springer-Verlag.

Damien Pous and Jana Wagemaker. Completeness theorems for kleene algebra with top. In Bartek Klin, Slawomir Lasota, and Anca Muscholl, editors, *33rd International Conference on Concurrency Theory (CONCUR 2022)*, volume 243 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:18, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

A.M Silva. *Kleene coalgebra*. s.n.] ; UB Nijmegen [host, S.l.; Nijmegen, 2010.

Steffen Smolka, Praveen Kumar, David M. Kahn, Nate Foster, Justin Hsu, Dexter Kozen, and Alexandra Silva. Scalable verification of probabilistic networks. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, page 190–203, Phoenix AZ USA, June 2019. ACM.

Cheng Zhang, Arthur Azevedo de Amorim, and Marco Gaboardi. On incorrectness logic and kleene algebra with top and tests. (arXiv:2108.07707), Aug 2022. arXiv:2108.07707 [cs].

Cheng Zhang, Arthur Azevedo de Amorim, and Marco Gaboardi. On incorrectness logic and kleene algebra with top and tests. *Proceedings of the ACM on Programming Languages*, 6(POPL):29:1–29:30, Jan 2022.

CURRICULUM VITAE

Joe Graduate

Basically, this needs to be worked out by each individual, however the same format, margins, typeface, and type size must be used as in the rest of the dissertation.