



Domain Reasoning in TopKAT

Cheng Zhang ✉ 

Boston University

Arthur Azevedo de Amorim ✉ 

Rochester Institute of Technology

Marco Gaboardi ✉ 

Boston University

Abstract

TopKAT is the algebraic theory of Kleene algebra with tests (KAT) extended with a top element. Compared to KAT, one pleasant feature of TopKAT is that, in relational models, the top element allows us to express the domain and codomain of a relation. This enables several applications in program logics, such as proving under-approximate specifications or reachability properties of imperative programs. However, while TopKAT inherits many pleasant features of KATs, such as having a decidable equational theory, it is incomplete with respect to relational models. In other words, there are properties that hold true of all relational TopKATs but cannot be proved with the axioms of TopKAT. This issue is potentially worrisome for program-logic applications, in which relational models play a key role.

In this paper, we further investigate the completeness properties of TopKAT with respect to relational models. We show that TopKAT is complete with respect to (co)domain comparison of KAT terms, but incomplete when comparing the (co)domain of arbitrary TopKAT terms. Since the encoding of under-approximate specifications in TopKAT hinges on this type of formula, the aforementioned incompleteness results have a limited impact when using TopKAT to reason about such specifications.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory; Theory of computation → Programming logic

Keywords and phrases Kleene algebra, Kleene Algebra With Tests, Kleene Algebra With Domain, Kleene Algebra With Top and Tests, Completeness, Decidability

Funding *Cheng Zhang*: National Science Foundation Grant No. 2040249 and No. 2314324.

Arthur Azevedo de Amorim: National Science Foundation Grant No. 2314323.

Marco Gaboardi: National Science Foundation Grant No. 2040249 and No. 2314324.

1 Introduction

Kleene algebra with tests (KAT) is an algebraic framework that extends Kleene algebra with an embedded Boolean algebra to model control structures like if-statement and while-loops [20]. This extension enables us to reason about several properties of imperative programs. For example, one of the key early results in the area was that KAT can encode Hoare logic, in the sense that any proof in the logic’s propositional fragment can be carried out faithfully using KAT equations [27, 21].

Some applications, however, require us to look beyond KAT. For example, Zhang et al. [41] recently proved that KAT alone cannot be used to encode incorrectness logic [31, 7]—a close cousin of Hoare logic with applications in bug finding [25, 36]. A similar result was proved by Struth [39], who showed that KAT cannot encode weakest liberal preconditions. If we view a program as a relation between its input and output states, both of these limitations arise from KAT’s lack of power to encode the (co)domain of a relation. Indeed, Möller et al. [30] proved that incorrectness logic could be encoded by extending KAT with a codomain operation. Independently, Zhang et al. provided a similar encoding [41] by extending KAT

with a top element, which can be used to express inequalities between codomains. They dubbed the resulting algebraic structure a TopKAT.

The present paper investigates the expressive power of TopKAT as a tool for (co)domain reasoning. As noted by Zhang et al. [41], one limitation of TopKAT is that it is not expressive enough to derive all valid equations between relations. More precisely, Zhang et al.'s encoding of incorrectness logic interprets the top element of the algebra as the complete relation, which relates all pairs of program states. Under this interpretation, the inequality $p \top p \geq p$ is valid, but unprovable using the theory of TopKAT [41]. This is a potential issue when using TopKAT to reason in incorrectness logic: though Zhang et al.'s encoding covers all the rules of propositional incorrectness logic, there could be inequalities about (co)domain that fall outside this fragment and cannot be established solely by the theory of TopKAT.

Pous et al. [34, 35] were able to make some progress on the issue, by showing we can obtain a complete axiomatic system for relational models TopKATs by adding in the inequality $p \top p \geq p$ as an additional axiom. In this paper, we look at the question from a different angle, instead of working with a more complex theory, we show that the original theory of TopKAT is complete with respect to relational models for (co)domain comparisons, namely the inequalities of the form $\top t_1 \geq \top t_2$ or $t_1 \top \geq t_2 \top$ where t_1, t_2 are KAT terms. Since these inequalities suffice to encode incorrectness logic, this completeness result lays a solid foundation for encoding program logics in TopKAT. We have also showed that this completeness result is tight, in the sense that it does not extend to the case where t_1 and t_2 contain the top element, by explicitly constructing two TopKAT terms that witness the incompleteness.

The result above is enabled by the homomorphic structure of the reduction [41, 33] from TopKAT to KAT. This discovery also let us shorten the proofs of previous results [41], and enables systematic generation of TopKAT complete interpretations from complete interpretations of KAT. We believe that this new representation of the reduction technique could also be of independent interest.

Structure of this paper and contributions: In Section 2, we present several previous results on KAT and TopKAT. Inspired by universal algebra [4], we characterize fundamental concepts, like interpretation and completeness, using homomorphisms. In Section 3, we uncover additional structure of the reduction technique [24, 33] in the case of TopKAT: the reduction from TopKAT to KAT is a TopKAT homomorphism. This discovery not only allows us to simplify several previous results [41] by avoiding tedious induction proofs; but also enables the techniques used in the later section. Section 4 presents the completeness results of TopKAT with respect (co)domain comparison. The codomain completeness result is proven by an equality that connects codomain operation with the language interpretation, and the domain completeness is then proven by applying the codomain completeness result to the opposite TopKAT.

2 Preliminaries

2.1 Extensions of Kleene algebra And Their Models

A *Kleene algebra* is an idempotent semiring with a star operation, written p^* , that satisfies the following *unfolding*, *left induction*, and *right induction* rules:

$$p^* = 1 + pp^* = 1 + p^*p, \quad pr + q \leq r \implies p^*q \leq r, \quad rp + q \leq r \implies qp^* \leq r;$$

the ordering here is the conventional ordering in idempotent semirings: $p \leq q \triangleq p + q = q$. It is known that the right-hand version of unfolding and induction rule can be removed while

preserving the same equational theory [23]. Yet, we will focus on the standard definition of KA in this paper.

► **Lemma 1.** *Following are well-known facts in Kleene algebra*

- *All the Kleene algebra operations preserve order.*
- *The following equations are true for the star operation:*

$$p^* \cdot p^* = p^* \qquad (p^*)^* = p^*.$$

A Kleene algebra with tests (KAT) is a Kleene algebra with an embedded Boolean algebra, where the conjunction, disjunction, and identities in the Boolean algebra coincide with the addition, multiplication, and the identities of Kleene algebra. We refer to elements of this embedded Boolean algebra as *tests*.

Given an algebraic theory, we can construct its *free model* over a finite set Σ , called the *alphabet* [4]. The free model consists of all the terms formed by Σ modulo provable equivalences of the algebra. The operations of the free model are obtained by lifting the term-level operations to equivalence classes.

The above construction can be extended to the case of KAT and TopKAT, suppose that we are given two disjoint finite sets K (the *action alphabet*) and B (the *test alphabet*). Elements of K and B are called *primitive actions* and *primitive tests*, respectively. KAT terms over the alphabet K, B are defined with the following grammar:

$$t \triangleq b \in B \mid p \in K \mid 1 \mid 0 \mid t_1 + t_2 \mid t_1 \cdot t_2 \mid t^* \mid \bar{t}_b,$$

where t_b does not contain primitive actions. The *free KAT* over K, B , written $\text{KAT}_{K,B}$, consists of terms over K, B modulo provable KAT equivalences. The tests of the free KAT are Boolean terms, i.e. terms formed by primitive tests and Boolean operations modulo Boolean axioms. A similar construction applies to TopKAT, where an additional symbol \top was added as the largest element in the theory; we denote the free TopKAT over K, B as $\text{TopKAT}_{K,B}$. We sometimes omit the alphabets K and B when they are irrelevant or can be inferred.

In the paper, we frequently consider terms modulo provable equalities, i.e. in the context of its corresponding free model. For example, given $t_1, t_2 \in \text{KAT}$, we will say $t_1 = t_2$ when they are provably equal using the theory of KAT. Although the free model seems trivial, it leads to simpler and more modular proofs of some properties of algebraic theories, as we will see in Section 3.

Other important models that we will use in this paper are language (Top)KATs and relational (Top)KATs, which we review here. An *atom* (short for “atomic test”) over a test alphabet $B = \{b_1, b_2, \dots, b_n\}$ is a sequence of the form

$$\hat{b}_1 \cdot \hat{b}_2 \cdot \dots \cdot \hat{b}_n \text{ where } \hat{b}_i \in \{b_i, \bar{b}_i\}.$$

We denote atoms as $\alpha, \beta, \gamma, \dots$ and the set of all atoms as At .

A *guarded string* (or *guarded word*) over K, B is an alternation between atoms and primitive actions that starts and ends in atoms:

$$\alpha_0 p_1 \alpha_1 \dots p_n \alpha_n \text{ where } p_i \in K, \alpha_i \in \text{At};$$

where each action is “guarded” by an atom. A guarded string is similar to a program trace, where each program state is denoted by an atom; and primitive actions will cause a transition between program states. We denote the set of all guarded strings over alphabet K, B as

$GS_{K,B}$, and we will omit the alphabet K, B when it is irrelevant or can be inferred from context. The notation αs denotes a guarded string starting with atom α with the rest of the string s ; similarly, $s\alpha$ denotes a guarded string that ends with atom α with rest of the string being s .

► **Definition 2** (Language/trace KAT [24]). *The language KAT (also called “trace KAT”) over an alphabet K, B is denoted as $\mathcal{G}_{K,B}$, or simply \mathcal{G} if no confusion can arise.*

The elements are sets of guarded strings (called guarded languages), and the tests are sets of atoms. The additive identity 0 is the empty set, and the multiplicative identity 1 is the set of all the atoms At . The addition operator is set union, and the multiplication operator is defined as follows:

$$S_1 \diamond S_2 \triangleq \{s_1 \alpha s_2 \mid s_1 \alpha \in S_1, \alpha s_2 \in S_2\}.$$

The star operation is defined non-deterministically iterating the multiplication operator:

$$S^* \triangleq \bigcup_{i \in \mathbb{N}} S^i \text{ where } S^0 = \text{At}, S^{k+1} = S \diamond S^k.$$

Another useful type of KAT are relational ones, where each element is a relation $R \subseteq X \times X$ over a fixed set X . In applications, the set X typically represents the set of all possible program states, and each relation R represents a program by relating each possible input to the corresponding output.

► **Definition 3** (Relational KAT). *A relational KAT is a KAT \mathcal{R} consists of relations over a fixed set X (though \mathcal{R} need not contain every relation over X), and it is closed under the following operations. The tests are all the relations that are subsets of the identity relation. The additive identity 0 is the empty set, and the multiplicative identity is the identity relation:*

$$1 \triangleq \{(x, x) \mid x \in X\}.$$

The addition operator is set union, and the multiplication operation is relational composition:

$$R_1; R_2 = \{(x, z) \mid \exists y \in X, (x, y) \in R_1, (y, z) \in R_2\}.$$

Finally, the star operation is defined as:

$$R^* \triangleq \bigcup_{i \in \mathbb{N}} R^i \text{ where } R^0 = 1, R^{k+1} = R; R^k.$$

We denote the class of all relational KATs as REL .

TopKAT extends the theory of KAT with the largest element \top , i.e. $\top \geq p$ for all elements p . The *language TopKAT* over an alphabet K, B has the same carrier and operations as $\mathcal{G}_{K \top, B}$, where $K \top$ is the set K joined with a new primitive action \top ; and the largest element is the full language $GS_{K \top, B}$.

The *relational TopKAT* is a relational KAT that contains the complete relation:

$$\top \triangleq \{(x, y) \mid x, y \in X\};$$

we denote the set of all relational TopKATs as TopREL . It is known that there are equations that are valid in relational TopKAT, but are not derivable by the axioms of TopKAT [41]; however, by adding the axiom $p \top p \geq p$, the theory becomes complete over relational TopKATs [34, 35]. In this paper, instead of working with a more complex theory, we will show

that TopKAT without any additional axiom already suffices for the purpose of encoding domain comparisons. Indeed, TopKAT is complete with respect to domain comparison inequalities, which can be used to encode both incorrectness logic and Hoare logic.

In this paper, we will use dom and cod to denote the conventional (co)domain operators on relations, namely, for any relation R :

$$\text{dom}(R) \triangleq \{x \mid \exists y, (x, y) \in R\} \quad \text{cod}(R) \triangleq \{y \mid \exists x, (x, y) \in R\}.$$

To demonstrate how TopKAT models (co)domain comparisons, we take any relational TopKAT \mathcal{R} and two relations $R_1, R_2 \in \mathcal{R}$, and we denote the complete relation as \top :

► **Lemma 4** (TopKAT encodes (co)domain comparison).

$$R_1 \top \supseteq R_2 \top \iff \text{dom}(R_1) \supseteq \text{dom}(R_2) \quad \top R_1 \supseteq \top R_2 \iff \text{cod}(R_1) \supseteq \text{cod}(R_2)$$

If we regard R_1 and R_2 as the input output relation of two programs, which is typically encoded by KAT terms, we can see that $R_1 \top \supseteq R_2 \top$ reflects that the domain of R_1 is larger than the domain of R_2 ; and similarly for the inequality $\top R_1 \supseteq \top R_2$. Thus, given two KAT terms $t_1, t_2 \in \text{KAT}_{K,B}$, we call inequalities like $t_1 \top \geq t_2 \top$ *domain comparison inequalities*, and $\top t_1 \geq \top t_2$ *codomain comparison inequalities*. Notice that the term $\top t_1$ is a shorthand for $\top \cdot i(t_1)$, where i is the inclusion function $\text{KAT}_{K,B} \hookrightarrow \text{TopKAT}_{K,B}$. In the rest of the paper, we will sometimes leave this inclusion function implicit. These two forms of inequalities will be the focus of our completeness results in Section 4.

We also know another class of TopKATs named *general relational TopKATs*, which is denoted as TopGREL. The top element of general relational TopKAT is not necessarily the complete relation, but the largest relation in the model. All equations in the general relational TopKAT can be derived using the theory of TopKAT.

However, the completeness of TopGREL came at the cost of expressive power: every predicate that is expressible using general relational TopKAT is already expressible using relational KAT [41], so the extension with top, in the case of general relational TopKAT, does not grant any extra expressive power. In Theorem 13, we show that this result is a simple corollary of our new reduction result.

We are also interested in maps between models: A *KAT homomorphism* f is a map between two KATs \mathcal{K} and \mathcal{K}' s.t. it preserves the sorts and operations: given a test b in \mathcal{K} then $f(b)$ is a test in \mathcal{K}' ; and all the KAT operations (complement, identities, addition, multiplication, and star) are preserved:

$$\begin{aligned} f &: \mathcal{K} \rightarrow \mathcal{K}' \\ f(\bar{b}) &= \overline{f(b)} \\ f(1) &= 1 \\ f(0) &= 0 \\ f(p + q) &= f(p) + f(q) \\ f(p \cdot q) &= f(p) \cdot f(q) \\ f(p^*) &= f(p)^*. \end{aligned}$$

Similarly, a *TopKAT homomorphism* is a KAT homomorphism that preserves the largest element.

2.2 Interpretation, Completeness, and Injectivity

Consider a KAT equation such as $p \cdot b \cdot \bar{b} = 0$. To determine its validity in a particular KAT \mathcal{K} , we need to assign meaning to it by interpreting each primitive as an element in \mathcal{K} ; that is, by defining a map \hat{I} of type $K + B \rightarrow \mathcal{K}$. Such a map $\hat{I} : K + B \rightarrow \mathcal{K}$ induces a unique KAT homomorphism $I : \text{KAT}_{K,B} \rightarrow \mathcal{K}$ inductively defined on the term as follows:

$$\begin{aligned}
 I(p) &\triangleq \hat{I}(p) && \text{where } p \in K + B \\
 I(\bar{t}_b) &\triangleq \overline{I(t_b)} && t_b \text{ does not contain primitive actions} \\
 I(t_1 + t_2) &\triangleq I(t_1) + I(t_2) \\
 I(t_1 \cdot t_2) &\triangleq I(t_1) \cdot I(t_2) \\
 I(t^*) &\triangleq I(t)^*
 \end{aligned} \tag{1}$$

In fact, every KAT homomorphism from a free model arises this way: there is a bijection between functions of type $K + B \rightarrow \mathcal{K}$ and KAT homomorphisms of type $\text{KAT}_{K,B} \rightarrow \mathcal{K}$, for any KAT \mathcal{K} . Because the homomorphism I and the function \hat{I} are equivalent, we will refer to them interchangeably as *KAT interpretations* and denote both of them as I .

The above result enables us to define a homomorphism from the free KAT just by defining its action on the primitives; saving us time to check the equations that a homomorphism must satisfy. It also allows us to prove that two interpretations are equal by arguing that they map the primitives to equal values.

Given a KAT \mathcal{K} , and two terms $t_1, t_2 \in \text{KAT}_{K,B}$ we say that $\mathcal{K} \models t_1 = t_2$ if

$$\forall I : \text{KAT}_{K,B} \rightarrow \mathcal{K}, I(t_1) = I(t_2).$$

In particular, for two terms in the free model $t_1, t_2 \in \text{KAT}_{K,B}$, $\text{KAT}_{K,B} \models t_1 = t_2$ is equivalent to $t_1 = t_2$. For a collection of models \mathbf{K} , we say that $\mathbf{K} \models t_1 = t_2$ if for all $\mathcal{K} \in \mathbf{K}$, $\mathcal{K} \models t_1 = t_2$. For example, $\text{REL} \models t_1 = t_2$ means that $t_1 = t_2$ is valid in all relational KATs. All the above notations and terminologies can be similarly extended to TopKAT.

Theories like KAT and TopKAT are designed to model practical programs, so it is important to know if they can model all the desirable equations between programs. If the theory of KAT can derive all the equalities for a particular interpretation I , namely:

$$\text{KAT}_{K,B} \models t_1 = t_2 \iff I(t_1) = I(t_2),$$

we say that the theory of KAT is *complete* with respect to I . Recall that $\text{KAT}_{K,B} \models t_1 = t_2$ is equivalent to $t_1 = t_2$; thus, by definition, an interpretation I is complete if and only if it is injective. One of such interpretation is the guarded string interpretation $G : \text{KAT}_{K,B} \rightarrow \mathcal{G}_{K,B}$ [24], defined by lifting the following action on the primitives:

$$G(b) = \{\alpha \mid b \text{ appears positively in } \alpha\}, \quad G(p) = \{\alpha p \beta \mid \alpha, \beta \in \text{At}\}.$$

In several previous works, the term “free model” refers to the range (set of reachable elements) of a complete interpretation. Since a complete interpretation is an injective homomorphism, such interpretation induces an isomorphism on its range, thus our definition of free model is equivalent to these definitions.

Many previous proofs can also be explained by seeing complete interpretations as injective homomorphisms: the proof for completeness of relational KATs constructs an injective homomorphism h from a language KAT into a relational KAT [24]. Since both G and h

are injective homomorphisms, $h \circ G$ is also an injective homomorphism, hence a complete interpretation. Since $h \circ G$ is a relational interpretation:

$$\text{KAT}_{K,B} \models t_1 = t_2 \implies \text{REL} \models t_1 = t_2 \implies h \circ G(t_1) = h \circ G(t_2);$$

then the completeness of $h \circ G$ implies $(h \circ G)(t_1) = (h \circ G)(t_2) \iff \text{KAT}_{K,B} \models t_1 = t_2$. Hence,

$$\text{KAT}_{K,B} \models t_1 = t_2 \iff \text{REL} \models t_1 = t_2,$$

i.e. the theory of KAT is complete with respect to relational KAT.

Besides using composition of injective homomorphisms, another technique commonly used to prove injectivity is to construct a left inverse: if a (Top)KAT homomorphism $f : \mathcal{K} \rightarrow \mathcal{K}'$ has a left inverse homomorphism $g : \mathcal{K}' \rightarrow \mathcal{K}$ i.e. $g \circ f = \text{id}_{\mathcal{K}}$, then f is injective. Notice that g does not need to be a homomorphism for f to be injective, however, in the case where f is an interpretation, g being a homomorphism makes the equality $g \circ f = \text{id}_{\mathcal{K}}$ easier to check. Because both $g \circ f$ and $\text{id}_{\mathcal{K}}$ are all interpretations, they are equal if and only if they have the same action on all the primitives.

Finally, we provide a shorthand for domain reasoning. For two terms $t_1, t_2 \in \text{KAT}$, we write

$$\text{REL} \models \text{dom}(t_1) \geq \text{dom}(t_2),$$

when $\text{dom}(I(t_1)) \supseteq \text{dom}(I(t_2))$ for all relational KAT interpretations I ; and similarly for relational TopKAT and general relational TopKAT. Then Lemma 4 implies the following:

► **Lemma 5.** *For two KAT terms $t_1, t_2 \in \text{KAT}_{K,B}$:*

$$\text{TopREL} \models t_1 \top \geq t_2 \top \iff \text{REL} \models \text{dom}(t_1) \geq \text{dom}(t_2)$$

$$\text{TopREL} \models \top t_1 \geq \top t_2 \iff \text{REL} \models \text{cod}(t_1) \geq \text{cod}(t_2)$$

3 Reduction, A New Perspective

Our goal in this section is to construct a complete interpretation for TopKAT, by reducing its theory to 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 [41, 42, 34], we present a more streamlined proof that hinges on the universal properties of free KATs and TopKATs, without relying explicitly on language models. Similar to previous works, we obtain the decidability of the equational theory of TopKAT as a corollary of reduction. However, because of the new notion of reduction, our decidability result no longer depends on the completeness of the language TopKAT. 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 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.

► **Lemma 6.** *Every free KAT over alphabet K, B forms a TopKAT.*

Proof. Since $\text{KAT}_{K,B}$ is a KAT, we only need to show the term $(\sum K)^*$ is the largest element of $\text{KAT}_{K,B}$, i.e.

$$(\sum K)^* \geq t, \forall t \in \text{KAT}_{K,B}.$$

The above fact can be shown by induction on t ; some algebraic manipulations below use facts in Lemma 1:

- $(\sum K)^* \geq 1$ (by unfolding rule), thus $(\sum K)^*$ is larger than 0, 1 and every Boolean term.
- $(\sum K)^*$ is larger than $\sum K$, which is larger than every primitive action.
- Given two terms t_1 and t_2 , assume $(\sum K)^*$ is larger than both. Because $(\sum K)^* = (\sum K)^* + (\sum K)^*$ and addition preserves order,

$$(\sum K)^* = (\sum K)^* + (\sum K)^* \geq t_1 + t_2$$

- Given two terms t_1 and t_2 , assume $(\sum K)^*$ is larger than both. Because $(\sum K)^* = (\sum K)^* \cdot (\sum K)^*$ and multiplication preserves order,

$$(\sum K)^* = (\sum K)^* \cdot (\sum K)^* \geq t_1 \cdot t_2.$$

- Given a term t , if $(\sum K)^* \geq t$, then $(\sum K)^* \geq t^*$. Since $(\sum K)^* = ((\sum K)^*)^*$ and star preserves order:

$$(\sum K)^* = ((\sum K)^*)^* \geq t^*. \quad \blacktriangleleft$$

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

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

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

In fact, such an injective homomorphism can be obtained by lifting the embedding map $K + B \hookrightarrow \text{KAT}_{K\top,B}$:

$$r : K + B \rightarrow \text{KAT}_{K\top,B}$$

$$r(p) \triangleq p.$$

This homomorphism coincides with the *reduction maps* of the same name in previous works [41, 35]. More concretely, we can picture r as simply replacing the symbol \top in a TopKAT term with $(\sum K\top)^*$, the largest element in $\text{KAT}_{K\top,B}$.

We will show that r is injective by constructing a left inverse for it. In fact, the left inverse $[-]_{\top}$ simply interprets the \top primitive in $\text{KAT}_{K\top,B}$ as the largest element.

► **Lemma 7.** *The map $[-]_{\top} : \text{KAT}_{K\top,B} \rightarrow \text{TopKAT}_{K,B}$, where each term is mapped to its corresponding equivalence class, is defined by lifting the following action on the primitives:*

$$\begin{aligned} [p]_{\top} &\triangleq p & \text{if } p \in K + B \\ [\top]_{\top} &\triangleq \top. \end{aligned}$$

The map $[-]_{\top}$ is a TopKAT homomorphism.

Proof. Because this map defined by lifting on the primitives, it is automatically a KAT homomorphism. All we need to show is that $[-]_{\top}$ preserves the top element, that is $[(\sum K_{\top})^*]_{\top} = (\sum K_{\top})^*$ is the largest element in $\text{TopKAT}_{K,B}$.

By construction of $\text{TopKAT}_{K,B}$, \top is the largest element in $\text{TopKAT}_{K,B}$. Thus, to prove that $(\sum K_{\top})^*$ is also the largest element in $\text{TopKAT}_{K,B}$, it suffices to prove $(\sum K_{\top})^* \geq \top$:

$$(\sum K_{\top})^* \geq \sum K_{\top} = \top + \sum K \geq \top. \quad \blacktriangleleft$$

► **Theorem 8 (Reduction).** $[-]_{\top}$ is the right inverse of r : $[-]_{\top} \circ r = \text{id}_{\text{TopKAT}_{K,B}}$. More explicitly for all $t \in \text{TopKAT}_{K,B}$:

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

Proof. Since $[-]_{\top} \circ r : \text{TopKAT}_{K,B} \rightarrow \text{TopKAT}_{K,B}$ is a TopKAT interpretation, the action on the primitives uniquely determines the interpretation: because both r and $[-]_{\top}$ are identity on the primitives, therefore $[-]_{\top} \circ r$ is the identity interpretation on $\text{TopKAT}_{K,B}$. ◀

The above theorem matches one of the soundness condition of reductions in previous works [41, 24, 33], which was typically proven by a monolithic induction on the structure of terms. Our approach, on the other hand, relies on establishing fine-grained algebraic properties, like Lemmas 6 and 7; then the theorem follows simply by computing the action of $[-]_{\top} \circ r$ on primitives.

Since r has a right inverse, it is indeed the injective interpretation we desired, and it is also a complete interpretation:

$$\text{TopKAT}_{K,B} \models t_1 = t_2 \iff r(t_1) = r(t_2),$$

With the completeness of r , we can already show the complexity of TopKAT . The complexity results echo previous proofs [41, 35], but we are able to obtain this result without completeness of TopKAT language interpretation, which is essential in previous proofs.

► **Corollary 9 (Complexity).** Given two terms $t_1, t_2 \in \text{TopKAT}_{K,B}$, deciding whether these two terms are equal is PSPACE-complete.

Proof. Deciding KAT equality is a sub-problem of deciding TopKAT equality, and KAT equality is PSPACE-hard [6]; therefore TopKAT equality is PSPACE-hard.

To decide the equality of t_1, t_2 , we first remove all the redundant primitives that do 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 [6] to decide whether $r(t_1) = r(t_2)$ in $\text{KAT}_{K_{\top}, B}$, this will also take polynomial space. Hence, the decision procedure for TopKAT equality is in PSPACE.

Thus deciding TopKAT equality is PSPACE-complete. ◀

3.2 Computing the complete interpretations

Designing complete interpretations and models was not always easy. In fact, in previous works [42], the authors made a mistake in the definition of language TopKAT , which was fixed later [41] by suggestion of Pous et al. [34]. However, with the results in Section 3.1, 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:

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

with a complete language model $\mathcal{G}_{K_\top,B}$, and a complete model consisting 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 does not change the multiplication operation, the multiplication in the language TopKAT is the same as in language KAT. In fact, as r does not change any operation in KAT, most operations in language TopKAT are the same as language KAT. Thus, we only need to compute 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 10.** *The language TopKAT inherits all the operations in language KAT, except the top element, which is defined as the full language. And such models are complete with $G \circ r$ as a complete interpretation.*

In the same way, we know that complete models consisting of relations (a.k.a. general relational TopKAT) will have the same operations as relational KATs. However, in this case the characterization of 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 11.** *The general relational TopKAT inherits all the operations in relational KAT, except the top element is the largest relation. And such models are 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 [41]; 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 does not give extra expressive power in general relational TopKAT.

The original proof [41, Lemma 2] encodes every TopKAT term using a KAT term, and then uses two pages to prove the soundness of this encoding. Here we show the aforementioned encoding is simply the reduction r .

► **Definition 12.** *Given two terms $t_1, t_2 \in \text{TopKAT}$, and n primitives $p_1, p_2, \dots, p_n \in K + B$, we say that an n -ary predicate P is expressible by equation $t_1 = t_2$ for a class of TopKATs \mathbf{K} when for all interpretations I into TopKATs in \mathbf{K} , the following equivalence holds:*

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

► **Theorem 13** (Expressiveness of general relational TopKAT). *Given an alphabet K, B , an n -ary predicate P on relations, the predicate P over primitives $p_1, p_2, \dots, p_n \in K$ is expressible in general relational TopKAT if and only if it is expressible in relational KAT.*

Proof. A predicate expressible in relational KAT is also expressible in general relational TopKAT using the same pair of terms, we only need to show the converse. Assume a predicate P is expressible in general relational TopKAT, then there exists two TopKAT terms $t_1, t_2 \in \text{TopKAT}_{K,B}$ s.t. for all general relational TopKAT interpretations I_\top :

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

We take an arbitrary relational KAT interpretation I from $\text{KAT}_{K_\top, B}$. Notice $\mathbf{Im}(I)$, the range of I , is a relational KAT with the largest element $I((\sum K)^*)$, i.e. $\mathbf{Im}(I)$ is a general relational TopKAT. Because I is a KAT interpretation, it preserves all the KAT operations and the largest element. Hence, I is a TopKAT homomorphism from $\text{KAT}_{K_\top, B}$ to $\mathbf{Im}(I)$.

Then we can construct $I \circ r : \text{TopKAT}_{K,B} \rightarrow \mathbf{Im}(I)$, a general relational interpretation:

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

Thus the two KAT terms $r(t_1), r(t_2) \in \text{KAT}_{K_\top, B}$ also can express the predicate P . \blacktriangleleft

Since the image of I is not necessarily a relational TopKAT, where the top element is interpreted as the complete relation, the above trick does 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 [41].

4 (Co)domain Completeness

In general, TopKAT is not complete over relational models, which are crucial for applications in program logics [41]. 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 [35].

In this paper, we take a different approach than Pous et al. [35]: instead of extending the TopKAT framework, we will restrict the completeness result. In particular, the encoding of incorrectness logic and Hoare Logic in TopKAT [41] 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 $t_1, t_2 \in \text{KAT}_{K,B}$ (i.e. \top does not appear in t_1 and t_2)

$$\begin{aligned} \text{REL} \models \text{cod}(t_1) \geq \text{cod}(t_2) &\iff \text{TopKAT} \models \top t_1 \geq \top t_2 && \text{codomain completeness} \\ \text{REL} \models \text{dom}(t_1) \geq \text{dom}(t_2) &\iff \text{TopKAT} \models t_1 \top \geq t_2 \top && \text{domain complete} \end{aligned}$$

In this section, we prove that these equivalences hold, even without the additional axiom. 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 $\text{dom}(p \top p) \geq \text{dom}(p)$. However, $p \top p \top \geq p \top$ is not provable in TopKAT, because the inequality is not valid with the language interpretation. The incompleteness of codomain comparison can also be shown using the same example.

4.1 Codomain completeness

The core insight to prove the domain completeness result is to construct a specific relational interpretation $h \circ i \circ G$, where its codomain is equivalent to the complete TopKAT

interpretation $G \circ r$:

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

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 [24], which we will recall as follows:

$$h(L) = \{(s, s \diamond s') \mid s \in GS, s' \in L\}.$$

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.

► **Lemma 14.** *For any term $t \in \text{KAT}_{K,B}$,*

$$\text{cod}(h \circ i \circ G(t)) = G \circ r(\top t).$$

Proof. We explicitly write out the domain and codomain of the functions in the relational KAT interpretation $h \circ i \circ G$ for the ease of the reader:

$$\text{KAT}_{K,B} \xrightarrow{G} \mathcal{G}_{K,B} \xrightarrow{i} \mathcal{G}_{K\top,B} \xrightarrow{h} \mathcal{P}(\mathcal{G}_{K\top,B} \times \mathcal{G}_{K\top,B}).$$

In this case, h is a KAT homomorphism from $\mathcal{G}_{K\top,B}$:

$$h(S) = \{(s, s \diamond s_1) \mid s \in GS_{K\top,B}, s_1 \in S\}.$$

Since the reduction r preserves terms without \top , let $t \in \text{KAT}_{K,B}$ (i.e. t does not contain \top),

$$G \circ r(\top) = GS_{K\top,B} \qquad G \circ r(t) = G(t).$$

Therefore, for any term $t \in \text{KAT}_{K,B}$

$$\begin{aligned} \text{cod}(h \circ i \circ G(t)) &= \{s\alpha s_1 \mid s\alpha \in GS_{K\top,B}, \alpha s_1 \in G(t)\} \\ &= GS_{K\top,B} \diamond G(t) \\ &= (G \circ r(\top)) \diamond (G \circ r(t)) \\ &= G \circ r(\top t). \end{aligned} \quad \blacktriangleleft$$

Lemma 14 established a connection between the codomain operator and the language interpretation of TopKAT. Then by completeness of the language interpretation, we will obtain the completeness of codomain comparison.

► **Theorem 15 (Codomain completeness).** *Given two terms $t_1, t_2 \in \text{KAT}_{K,B}$ (i.e. terms without \top), then codomain comparison is complete:*

$$\text{REL} \models \text{cod}(t_1) \geq \text{cod}(t_2) \iff \text{TopKAT} \models \top t_1 \geq \top t_2.$$

Proof. Given the natural inclusion homomorphism: $i : \text{KAT}_{K,B} \rightarrow \text{KAT}_{K\top,B}$, we show that the following are equivalent:

1. $\text{REL} \models \text{cod}(t_1) \geq \text{cod}(t_2)$.
2. $\text{cod}(h \circ i \circ G(t_1)) \geq \text{cod}(h \circ i \circ G(t_2))$.

3. $\text{TopKAT} \models \top t_1 \geq \top t_2$.

We first show that $1 \implies 2$, by definition, $\text{REL} \models \text{cod}(t_1) \geq \text{cod}(t_2)$ implies $\text{cod}(I(t_1)) \geq \text{cod}(I(t_2))$ for all relational KAT interpretations I . Because $h \circ i \circ G$ is a relational KAT interpretation, so $1 \implies 2$.

We show $2 \implies 3$, which uses the equality discussed above, and proved in Lemma 14:

$$\begin{aligned} & \text{cod}(h \circ i \circ G(t_1)) \geq \text{cod}(h \circ i \circ G(t_2)) \\ \iff & G \circ r(\top t_1) \geq G \circ r(\top t_2) && \text{Lemma 14} \\ \iff & \text{TopKAT} \models \top t_1 \geq \top t_2. && \text{Completeness of } G \circ r \end{aligned}$$

Finally, we show $3 \implies 1$, by Lemma 5:

$$\text{TopKAT} \models \top t_1 \geq \top t_2 \implies \text{TopREL} \models \top t_1 \geq \top t_2 \implies \text{REL} \models \text{cod}(t_1) \geq \text{cod}(t_2). \quad \blacktriangleleft$$

4.2 Domain completeness

The domain completeness result can be derived from codomain completeness by observing properties of opposite TopKAT and the converse operator $(-)^{\vee}$, both of which we will recall below.

For every TopKAT \mathcal{K} , we can construct the opposite TopKAT \mathcal{K}^{op} by reversing the multiplication operation, keeping the sorts and other operations unchanged:

$$p \hat{\cdot} q \triangleq q \cdot p,$$

where $\hat{\cdot}$ is multiplication in \mathcal{K}^{op} and \cdot is multiplication in \mathcal{K} . By definition, $(-)^{\text{op}}$ is an involution, that is $(\mathcal{K}^{\text{op}})^{\text{op}} = \mathcal{K}$. Furthermore, $(-)^{\text{op}}$ is a TopKAT functor, this means all TopKAT homomorphisms $h : \mathcal{K} \rightarrow \mathcal{K}'$ can be lifted to a TopKAT homomorphism on the opposite TopKAT $h^{\text{op}} : \mathcal{K}^{\text{op}} \rightarrow \mathcal{K}'^{\text{op}}$. The lifting h^{op} is point-wise equal to h :

$$\forall p \in \mathcal{K}, h^{\text{op}}(p) \triangleq h(p).$$

The fact that h^{op} is a TopKAT homomorphism can be proven by unfolding the definition, and the functor laws are satisfied because h^{op} is point-wise equal to h .

There are two important homomorphisms involving opposite TopKAT:

$$\begin{aligned} (-)^{\vee} : (X \times X)^{\text{op}} &\rightarrow (X \times X) && \text{op} : \text{TopKAT}_{K,B} \rightarrow \text{TopKAT}_{K,B}^{\text{op}} \\ (R)^{\vee} &= \{(b, a) \mid (a, b) \in R\}, && \forall p \in K + B, \text{op}(p) = p. \end{aligned}$$

The $(-)^{\vee}$ is the relational converse operator, the rules of homomorphism can simply be proven by unfolding of definitions. The crucial property of $(-)^{\vee}$ is that it flips the domain and codomain:

$$\text{dom}(R^{\vee}) = \text{cod}(R). \quad (2)$$

Hence, allowing us to flip the result about codomains and apply it to domains.

op is a homomorphism from free TopKAT to its opposite TopKAT; it can be defined by lifting the embedding function $K + B \hookrightarrow \text{TopKAT}_{K,B}$ on primitives. Intuitively, given a term $t \in \text{TopKAT}$, $\text{op}(t)$ will flip all the multiplications in t recursively.

► **Lemma 16.** *the left inverse of op can be obtained by lifting itself through the $(-)^{\text{op}}$ functor,*

$$\text{op}^{\text{op}} : \text{TopKAT}^{\text{op}} \rightarrow (\text{TopKAT}^{\text{op}})^{\text{op}} = \text{TopKAT}.$$

Recall op^{op} is pointwise equal to op , thus $\text{op}^{\text{op}} \circ \text{op} : \text{TopKAT} \rightarrow \text{TopKAT}$ is the identity interpretation because it preserves all the primitives. Thus, op has a left inverse, hence it is injective:

$$t_1 = t_2 \iff \text{op}(t_1) = \text{op}(t_2).$$

Finally, since the elements in $\text{TopKAT}^{\text{op}}$ are the same as TopKAT , which are TopKAT terms modulo provable TopKAT equalities, theorems about TopKAT terms are also true for elements in $\text{TopKAT}^{\text{op}}$. In particular, codomain completeness (Theorem 15) also holds in $\text{TopKAT}^{\text{op}}$: for all terms $t_1, t_2 \in \text{TopKAT}$,

$$\top \cdot \text{op}(t_1) \geq \top \cdot \text{op}(t_2) \iff \text{REL} \models \text{cod}(\text{op}(t_1)) = \text{cod}(\text{op}(t_2)). \quad (3)$$

► **Theorem 17 (Domain Completeness).** *For all terms $t_1, t_2 \in \text{KAT}$, the following equivalence hold:*

$$\text{REL} \models \text{dom}(t_1) = \text{dom}(t_2) \iff \text{TopKAT} \models t_1 \top \geq t_2 \top.$$

Proof. \Leftarrow direction is trivial by Lemma 5; and \Rightarrow direction can be derived as follows: let I be some relational interpretation, then $I^{\text{op}}(\text{op}(-))^{\vee}$ is also a relational interpretation:

$$I^{\text{op}}(\text{op}(-))^{\vee} : \text{TopKAT} \xrightarrow{\text{op}} \text{TopKAT}^{\text{op}} \xrightarrow{I^{\text{op}}} (X \times X)^{\text{op}} \xrightarrow{(-)^{\vee}} (X \times X).$$

Thus, we let I range over all relational interpretations:

$$\begin{aligned} \text{REL} \models \text{dom}(t_1) &\supseteq \text{dom}(t_2) \\ \implies \forall I, \text{dom}(I(t_1)) &\supseteq \text{dom}(I(t_2)) \\ \implies \forall I, \text{dom}(I^{\text{op}}(\text{op}(t_1))^{\vee}) &\supseteq \text{dom}(I^{\text{op}}(\text{op}(t_2))^{\vee}) && \text{specialize } I \text{ as } I^{\text{op}}(\text{op}(-))^{\vee} \\ \implies \forall I, \text{cod}(I^{\text{op}}(\text{op}(t_1))) &\supseteq \text{cod}(I^{\text{op}}(\text{op}(t_2))) && \text{Equation (2)} \\ \implies \forall I, \text{cod}(I(\text{op}(t_1))) &\supseteq \text{cod}(I(\text{op}(t_2))) && I^{\text{op}} \text{ is pointwise equal to } I \\ \implies \top \cdot \text{op}(t_1) &\geq \top \cdot \text{op}(t_2) && \text{Equivalence (3)} \\ \implies \text{op}(\top \cdot t_1) &\geq \text{op}(\top \cdot t_2) && \text{Definition of op} \\ \implies t_1 \top &\geq t_2 \top && \text{Lemma 16} \end{aligned}$$

◀

► **Remark 18.** Alternatively, Theorem 17 can also be proven by constructing the following h' :

$$\begin{aligned} h' : \mathcal{G}_{K,B} &\rightarrow \mathcal{P}(\mathcal{G}_{K,B} \times \mathcal{G}_{K,B}) \\ h'(S_1) &\triangleq \{(s_1 \alpha s, \alpha s) \mid s_1 \alpha \in S_1, \alpha s \in GS_{K,B}\}. \end{aligned}$$

Then the proof would mirror that of Theorem 15, replacing h with h' and replacing cod with dom . However, the proof of Theorem 17 reveals more properties of maps like $(-)^{\vee}$ and op , thus we choose to present the current proof of Theorem 17 instead of the alternative proof.

5 Related Works

Extensions of Kleene algebra and reduction: soon after the completeness of Kleene algebra was proven [18], it was realized that adding an embedded Boolean algebra can help reasoning about control structures, such system is referred to as Kleene algebra with tests (KAT) [24, 6]. Later KAT was further extended to reason about failure [26], indicator variables [13], domain [9], networks [1], and relational reasoning [3]. Kleene algebra has also been extended to reason about concurrency, as concurrent Kleene algebra [14, 17] and concurrent Kleene algebra with observations [16]. Many of these extensions can be seen as Kleene algebra with extra hypotheses [5, 11]. Although many hypotheses make the theory undecidable [19, 22, 11], many useful hypotheses can be eliminated via reduction [33]. Thus, our new perspective on reduction could potentially lead to streamlining of various previous proofs, and more general proofs of completeness results.

Top element: Tarski's relational algebra [40] contains the addition, multiplication, and identity operation of KA; in addition, relational algebra also include a top element. Hence attempts to incorporate Kleene star into relational algebra effectively create a super theory of TopKAT. Unfortunately, several attempts at these algebras turn out to be undecidable because of the presence of intersection and converse operations [2, 32]. With the intersection and converse operators removed, top element is proven to be individually useful in Kleene algebra: for example, Mamouras [26] uses the top element to forget program states, and Antonopoulos et al. [3] uses top to design forward simulation rules for relational verification, and claim that relational incorrectness logic [29] can be encoded using BiKAT extended with top. The completeness and decidability of TopKAT was first studied by Zhang et al. [41], and concluded that TopKAT is not complete with relational models. Later, Pous et al. [34, 35] showed that both TopKA and TopKAT is complete with relational model with one additional axiom: $p \top p \geq p$, and the theory remains PSPACE-complete, like KAT and TopKAT. In this paper, we showed that TopKAT without the additional axiom is complete for a specific form of inequalities, namely when top only appears in the front or the end of the term. Although this form of inequalities seem restrictive, they are enough to encode both Hoare and incorrectness logic [41].

Domain in KAT: The study of axiomatizing (co)domain in KAT has a long and rich history. Domain semiring [10] and Kleene algebra with domain [9] were two popular yet different axiomatizations of (co)domain in Kleene algebra with tests. These two axiomatizations turn out to coincide in a large class of semirings [12]. Various applications for domain in KAT have been discovered, including modeling program correctness, predicate transformers, temporal logics, termination analysis, and many more [8]. Many of these applications can even be efficiently automated [15]. However, although the free relational model of these theories has been characterized [28], the search for general complete interpretation remains unfruitful. The complexity of these theories was recently shown to be EXPTIME-complete [37], a worse complexity class than PSPACE-complete for TopKAT.

6 Conclusion And Open Problems

In this paper, we exploit the homomorphic structure of reduction to simplify the proof of various previous results [41]. We have also showed that TopKAT is complete with respect to (co)domain comparison in the relational models, which lays a solid foundation for the use of TopKAT in (co)domain reasoning.

However, there are still several interesting unsolved problems about TopKAT. Most of

the incorrectness logic rules are written using hypotheses, for example, the sequencing rule:

$$\frac{[a] \ p \ [b] \quad [b] \ q \ [c]}{[a] \ p \cdot q \ [c]}$$

corresponds to the implication $\top a p \leq \top b \wedge \top b p \leq \top c \implies \top a p q \leq \top c$. Although each individual inequality in the implication fits the desired form $\top t_1 \geq \top t_2$, it is unclear whether implications of the form

$$\top t_{11} \leq \top t_{12} \wedge \top t_{21} \leq \top t_{22} \wedge \dots \wedge \top t_{n1} \leq \top t_{n2} \implies \top t_1 \leq \top t_2$$

are complete with relational TopKAT or decidable.

Recently, there is an efficient fragment of KAT proposed, named *Guarded Kleene algebra with tests* [38] or *GKAT*. This fragment not only enjoys nearly-linear time equality checking, but also soundly models probabilistic computations as well. It would be interesting to see whether the completeness and decidability result of TopKAT can be extended to GKAT, and whether the efficiency of GKAT will persist with the addition of top.

References

- 1 Carolyn Jane Anderson, Nate Foster, Arjun Guha, Jean-Baptiste Jeannin, Dexter Kozen, Cole Schlesinger, and David Walker. Netkat: semantic foundations for networks. *ACM SIGPLAN Notices*, 49(1):113–126, Jan 2014. doi:10.1145/2578855.2535862.
- 2 Hajnal Andr  ka and Szabolcs Mikul  s. Axiomatizability of positive algebras of binary relations. *Algebra universalis*, 66(1-2):7–34, October 2011. doi:10.1007/s00012-011-0142-3.
- 3 Timos Antonopoulos, Eric Koskinen, Ton Chanh Le, Ramana Nagasamudram, David A. Naumann, and Minh Ngo. An Algebra of Alignment for Relational Verification. *Proceedings of the ACM on Programming Languages*, 7(POPL):20:573–20:603, January 2023. doi:10.1145/3571213.
- 4 Stanley Burris and H. P. Sankappanavar. *A Course in Universal Algebra*. Number 78 in Graduate Texts in Mathematics. Springer-Verlag, New York, 1981.
- 5 Ernie Cohen. Hypotheses in kleene algebra. Mar 1995.
- 6 Ernie Cohen, Dexter Kozen, and Frederick Smith. The Complexity of Kleene Algebra with Tests. Technical Report, Cornell University, USA, June 1996.
- 7 Edsko de Vries and Vasileios Koutavas. Reverse Hoare Logic. In Gilles Barthe, Alberto Pardo, and Gerardo Schneider, editors, *Software Engineering and Formal Methods*, volume 7041, pages 155–171. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-24690-6_12.
- 8 Jules Desharnais, Bernhard M  ller, and Georg Struth. Modal kleene algebra and applications – a survey. In *Journal on Relational Methods in Computer Science*, page 93–131, 2004.
- 9 Jules Desharnais, Bernhard M  ller, and Georg Struth. Kleene algebra with domain. *ACM Transactions on Computational Logic*, 7(4):798–833, Oct 2006. doi:10.1145/1183278.1183285.
- 10 Jules Desharnais and Georg Struth. Internal axioms for domain semirings. *Science of Computer Programming*, 76(3):181–203, Mar 2011. doi:10.1016/j.scico.2010.05.007.
- 11 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.
- 12 Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemi  nski. Domain semirings united. *arXiv:2011.04704 [cs]*, Mar 2021.
- 13 Niels Bj  rn Bugge Grathwohl, Dexter Kozen, and Konstantinos Mamouras. Kat + b! In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer*

- Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, CSL-LICS '14, page 1–10, New York, NY, USA, Jul 2014. Association for Computing Machinery. doi:10.1145/2603088.2603095.
- 14 Tony Hoare, Stephan van Staden, Bernhard Möller, Georg Struth, and Huibiao Zhu. Developments in concurrent kleene algebra. *Journal of Logical and Algebraic Methods in Programming*, 85(4):617–636, Jun 2016. doi:10.1016/j.jlamp.2015.09.012.
 - 15 Peter Höfner and Georg Struth. Automated Reasoning in Kleene Algebra. In Frank Pfenning, editor, *Automated Deduction – CADE-21*, Lecture Notes in Computer Science, pages 279–294, Berlin, Heidelberg, 2007. Springer. doi:10.1007/978-3-540-73595-3_19.
 - 16 Tobias Kappé, Paul Brunet, Alexandra Silva, Jana Wagemaker, and Fabio Zanasi. *Concurrent Kleene Algebra with Observations: From Hypotheses to Completeness*, volume 12077 of *Lecture Notes in Computer Science*, page 381–400. Springer International Publishing, Cham, 2020. doi:10.1007/978-3-030-45231-5_20.
 - 17 Tobias Kappé, Paul Brunet, Alexandra Silva, and Fabio Zanasi. Concurrent kleene algebra: Free model and completeness. In Amal Ahmed, editor, *Programming Languages and Systems*, Lecture Notes in Computer Science, page 856–882, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-319-89884-1_30.
 - 18 D. Kozen. A completeness theorem for kleene algebras and the algebra of regular events. *Information and Computation*, 110(2):366–390, May 1994. doi:10.1006/inco.1994.1037.
 - 19 Dexter Kozen. *Kleene algebra with tests and commutativity conditions*, volume 1055 of *Lecture Notes in Computer Science*, page 14–33. Springer Berlin Heidelberg, Berlin, Heidelberg, 1996. doi:10.1007/3-540-61042-1_35.
 - 20 Dexter Kozen. Kleene algebra with tests. *ACM Transactions on Programming Languages and Systems*, 19(3):427–443, May 1997. doi:10.1145/256167.256195.
 - 21 Dexter Kozen. On hoare logic and kleene algebra with tests. *ACM Transactions on Computational Logic*, 1(1):60–76, Jul 2000. doi:10.1145/343369.343378.
 - 22 Dexter Kozen. On the complexity of reasoning in kleene algebra. *Information and Computation*, 179(2):152–162, Dec 2002. doi:10.1006/inco.2001.2960.
 - 23 Dexter Kozen and Alexandra Silva. Left-handed completeness. *Theoretical Computer Science*, 807:220–233, Feb 2020. doi:10.1016/j.tcs.2019.10.040.
 - 24 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. doi:10.1007/3-540-63172-0_43.
 - 25 Quang Loc Le, Azalea Raad, Jules Villard, Josh Berdine, Derek Dreyer, and Peter W. O’Hearn. Finding real bugs in big programs with incorrectness logic. *Proceedings of the ACM on Programming Languages*, 6(OOPSLA1):1–27, Apr 2022. doi:10.1145/3527325.
 - 26 Konstantinos Mamouras. *Equational Theories of Abnormal Termination Based on Kleene Algebra*, volume 10203 of *Lecture Notes in Computer Science*, page 88–105. Springer Berlin Heidelberg, Berlin, Heidelberg, 2017. doi:10.1007/978-3-662-54458-7_6.
 - 27 Ernest G. Manes and Michael A. Arbib. *Algebraic Approaches to Program Semantics*. Springer New York, New York, NY, 1986.
 - 28 Brett McLean. Free Kleene algebras with domain. *Journal of Logical and Algebraic Methods in Programming*, 117:100606, December 2020. doi:10.1016/j.jlamp.2020.100606.
 - 29 Toby Murray. An Under-Approximate Relational Logic. *Archive of Formal Proofs*, March 2020.
 - 30 Bernhard Möller, Peter O’Hearn, and Tony Hoare. *On Algebra of Program Correctness and Incorrectness*, volume 13027 of *Lecture Notes in Computer Science*, page 325–343. Springer International Publishing, Cham, 2021. doi:10.1007/978-3-030-88701-8_20.
 - 31 Peter W. O’Hearn. Incorrectness logic. *Proceedings of the ACM on Programming Languages*, 4(POPL):1–32, Jan 2020. doi:10.1145/3371078.

- 32 Damien Pous. On the Positive Calculus of Relations with Transitive Closure. In *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.STACS.2018.3.
- 33 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. doi:10.1007/978-3-030-88701-8_23.
- 34 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. doi:10.4230/LIPIcs.CONCUR.2022.26.
- 35 Damien Pous and Jana Wagemaker. Completeness theorems for kleene algebra with tests and top. (arXiv:2304.07190), Apr 2023.
- 36 Azalea Raad, Josh Berdine, Hoang-Hai Dang, Derek Dreyer, Peter O’Hearn, and Jules Villard. *Local Reasoning About the Presence of Bugs: Incorrectness Separation Logic*, volume 12225 of *Lecture Notes in Computer Science*, page 225–252. Springer International Publishing, Cham, 2020. doi:10.1007/978-3-030-53291-8_14.
- 37 Igor Sedlár. On the complexity of kleene algebra with domain. In *Relational and Algebraic Methods in Computer Science: 20th International Conference, RAMiCS 2023, Augsburg, Germany, April 3–6, 2023, Proceedings*, page 208–223, Berlin, Heidelberg, Apr 2023. Springer-Verlag. doi:10.1007/978-3-031-28083-2_13.
- 38 Steffen Smolka, Nate Foster, Justin Hsu, Tobias Kappé, Dexter Kozen, and Alexandra Silva. Guarded kleene algebra with tests: verification of uninterpreted programs in nearly linear time. *Proceedings of the ACM on Programming Languages*, 4(POPL):1–28, Jan 2020. doi:10.1145/3371129.
- 39 Georg Struth. On the expressive power of kleene algebra with domain. (arXiv:1507.07246), Aug 2015.
- 40 Alfred Tarski. On the calculus of relations. *Journal of Symbolic Logic*, 6(3):73–89, September 1941. doi:10.2307/2268577.
- 41 Cheng Zhang, Arthur Azevedo de Amorim, and Marco Gaboardi. On incorrectness logic and kleene algebra with top and tests. (arXiv:2108.07707), Aug 2022. doi:10.48550/arXiv.2108.07707.
- 42 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. doi:10.1145/3498690.