On Incorrectness Logic and Kleene Algebra with Top and Tests

CHENG ZHANG, ARTHUR AZEVEDO DE AMORIM, and MARCO GABOARDI, Boston University, USA

Kleene algebra with tests (KAT) is a foundational equational framework for reasoning about programs, which has found applications in program transformations, networking and compiler optimizations, among many other areas. In his seminal work, Kozen proved that KAT subsumes propositional Hoare logic, showing that one can reason about the (partial) correctness of while programs by means of the equational theory of KAT. In this work, we investigate the support that KAT provides for reasoning about *incorrectness*, instead, as embodied by O'Hearn's recently proposed incorrectness logic. We show that KAT cannot directly express incorrectness logic. The main reason for this limitation can be traced to the fact that KAT cannot express explicitly the notion of codomain, which is essential to express incorrectness triples. To address this issue, we study Kleene Algebra with Top and Tests (TopKAT), an extension of KAT with a top element. We show that TopKAT is powerful enough to express a codomain operation, to express incorrectness triples, and to prove all the rules of incorrectness logic sound. This shows that one can reason about the incorrectness of while-like programs by means of the equational theory of TopKAT.

CCS Concepts: • **Theory of computation → Program specifications**; **Pre- and post-conditions**; **Abstraction**.

Additional Key Words and Phrases: Program Reasoning, Incorrectness Logic, Hoare Logic, Kleene Algebra with Tests,

ACM Reference Format:
Cheng Zhang, Arthur Azevedo de Amorim, and Marco Gaboardi. 2022. On Incorrectness Logic and Kleene Algebra with Top and Tests. *Proc. ACM Program. Lang.* 6, POPL, Article 29 (January 2022), 30 pages. https://doi.org/10.1145/3498690

1 INTRODUCTION

Since the seminal work of Floyd [1967] and Hoare [1969], logic has become an essential tool for program verification. A program logic provides a system of *deduction rules* to prove *Hoare triples* $\{b\}\ p\ \{c\}$, where p is a program, and b and c are assertions describing the pre- and post-conditions. Such Hoare triples are (partial) correctness specifications, which state that, if we run p on an initial state satisfying b, all the final states (if reached) will satisfy c. In other words, c *over-approximates* the set of final states of p starting from b.

Despite their popularity, such logics are not the only tool for verifying programs. Another long-standing approach, which can be traced back to works by Taylor [1979] and O'Donnell [1985] in the 80s, is *equational reasoning*. Programs are modeled as elements of some algebraic structure, whose equational theory can be used to prove specifications. One such algebraic structure is *Kleene algebra with tests* [Kozen 1997] (KAT), which extends Kleene algebra with predicates for modeling conditionals. Kleene algebras with tests have several pleasing properties, such as equality

Authors' address: Cheng Zhang, czhang03@bu.edu; Arthur Azevedo de Amorim, arthur.aa@gmail.com; Marco Gaboardi, gaboardi@bu.edu, Department of Computer Science, Boston University, Boston, MA, 02215, USA.

2475-1421/2022/1-ART29
https://doi.org/10.1145/3498690

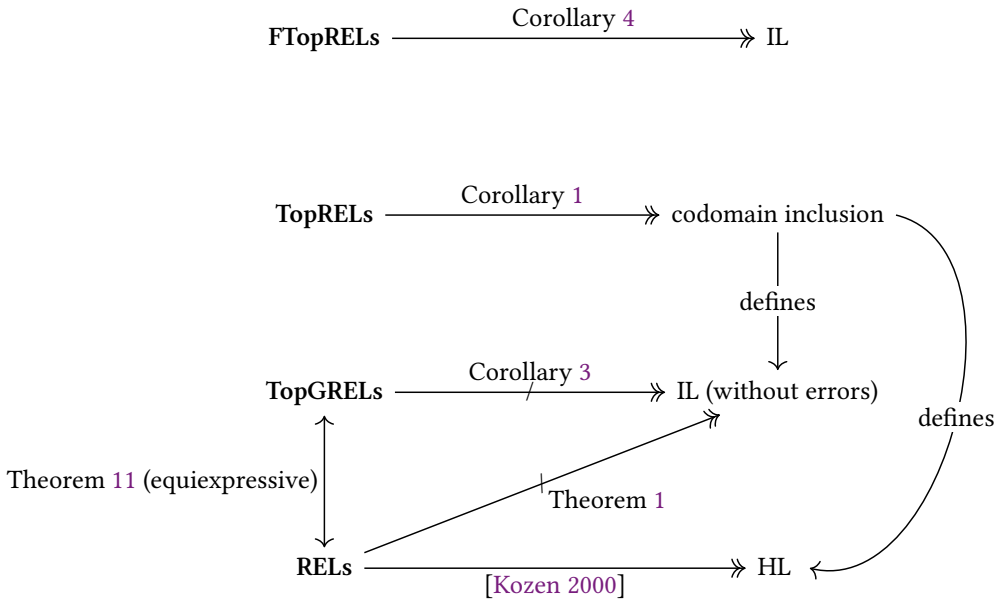

Fig. 1. Expressiveness of different systems.

of terms being decidable in PSPACE, and have been applied in several domains, including program transformations [Angus and Kozen 2001], networking [Anderson et al. 2014; Smolka et al. 2017], compiler optimization [Kozen and Patron 2000], and more.

The two approaches turned out to be not so different. Kozen [2000] showed that KAT can express the validity of a Hoare triple as an equation, in such a way that the deduction rules of a large fragment of Hoare Logic can be obtained by equational reasoning. Kozen's work established a clear bridge between deductive and equational reasoning for program verification, thus shedding light on the expressive power of KAT.

In this work, we are interested in extending this correspondence to other deductive formalisms for reasoning about programs. One such formalism is *incorrectness logic* (IL), a recent proposal by O'Hearn [2020] that relates to earlier works by de Vries and Koutavas [2011]. Instead of correctness, as in the Floyd-Hoare tradition, the system revolves around *incorrectness specifications* of the form $[b]\ p\ [c]$, which state that p can produce *any* final state satisfying c from *some* input state satisfying b—or, equivalently, that c *under-approximates* the set of final states of p starting from b. Though less conventional than traditional Hoare logic, incorrectness logic has already found its way to several applications, such as variants of separation logic [Raad et al. 2020] and relational verification for noninterference [Murray 2020].

It is natural to wonder whether Kozen's idea could be adapted to encode incorrectness logic in Kleene algebra with tests. Unfortunately, this is not the case. As we will show in this paper, there are incorrectness triples that cannot be expressed by any KAT equation. This might appear surprising, given the symmetry between over- and under-approximation in the formulation of Hoare logic and incorrectness logic. However, the symmetry involves the *image* of a set by a relation, an operation that is not part of the syntax of KAT. Several prior works have considered enlarging KAT with similar operations [Desharnais et al. 2004, 2006; Fahrenberg et al. 2021], but we show here that a smaller extension also serves our purposes: namely, adding a top element $\top$ to KAT. We call such a

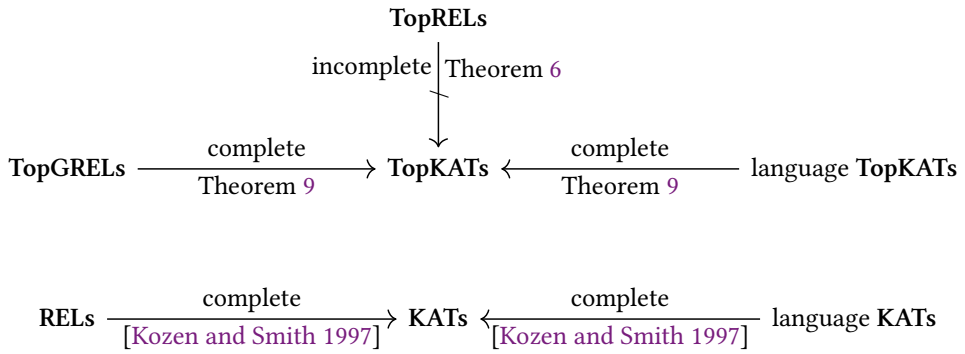

Fig. 2. Completeness relationships between classes of Kleene Algebras with Tests.

structure a *Kleene algebra with tests and top*, or a TopKAT. We show that such structures can encode inequalities between images, which we use to express incorrectness triples. The encoding allows us to prove the rules of incorrectness logic equationally, thus extending Kozen's correspondence to incorrectness reasoning. In fact, the use of TopKAT to encode incorrectness logic was also suggested by O'Hearn [2020].

For clarity of exposition, our main focus is on the fragment of incorrectness logic that handles normal program termination. However, O'Hearn [2020] also considered triples of the form $[p]\ c\ [er : q]$, whose interpretation is similar to the one we described above, except that they assume that execution can terminate with a fatal error (e.g. a failed assertion). Following Mamouras [2017], we show that our encoding carries over to such triples by considering FailTopKAT, an extension of TopKAT that includes an element `Fail` for representing failure. We prove that the abnormal termination rules of incorrectness logic follow from the equations of FailTopKAT.

We summarize our encodings in Figure 1. We use the notation $A \twoheadrightarrow B$ to mean that the logic B can be expressed in the equational theory of A. More formally, we model the ground-truth notion of validity in each logic as a statement about sets (assertions about program states) and relations (the input-output graph of a program). The encodings show that such statements are equivalent to equations involving operations in *relational* algebraic structures, where the carrier of the structure is some set of relations between program states (for example, **FTopRELs** is the class of relational FailTopKATs in Definition 21). Moreover, we prove that the equational theory alone (i.e., not specialized to relations) suffices to derive the rules of each logic. We also use the arrow $\rightarrow$ to denote the fact that the system which is able to express codomain can express both Hoare logic and incorrectness logic.

To evaluate the usefulness of these encodings, we investigate two basic properties of TopKAT: *completeness* and *decidability*. We say that an equational theory is complete for a certain class of structures if it can derive any equation that is valid in the class. We are particularly interested in completeness with respect to relational structures, since they are the natural setting for formulating program logics. However, It is well-known that the addition of a top element can be problematic for completeness of relational structures [Pous 2016], and we show that this is the case for TopKAT as well: the theory is incomplete for **TopRELs**, the class of relational structures where the top element is the complete relation. However, we do get completeness by considering a larger class **TopGRELs**, where the top element might not be the complete relation. We also show that TopKAT is complete for so-called *language TopKATs*, a class of structures inspired by prior work on KAT. (Figure 2 summarizes the relationships between these different structures.) Finally, we show that the equality

of TopKAT terms can be decided in PSPACE, by reducing a TopKAT term into a KAT term and use the PSPACE algorithm for KAT equalities [Cohen et al. 1999].

Summarizing, our contributions are:

- We show that (propositional) incorrectness logic cannot be encoded in relational KATs. Consequently, KAT cannot be used to reason equationally about incorrectness triples in general.
- We consider TopKAT, an extension of KAT with an additional top element $\top$, and show that (propositional) incorrectness logic for programs without error primitives can be encoded in relational TopKATs, by using $\top$ to encode the codomain of a relation. We prove that all the rules of this fragment of incorrectness logic can be derived solely by appealing to the equational theory of TopKAT.
- We consider FailTopKAT, an extension of TopKAT by means of a element `Fail` and we show that this can be used to encode incorrectness logic with an error primitive.
- We study the relations between the different systems we present in terms of expressivity and completeness.
- We prove that deciding equality of TopKAT terms are PSPACE-complete.

Several proofs are omitted in this paper to keep the paper concise, readers can find the omitted proofs in the full version [Zhang et al. 2021].

2 BACKGROUND

2.1 Klenee Algebra with Tests

Kleene algebra with tests was introduced by Kozen [1997] as an extension of Kleene algebra targeting program verification. The equational theory of Kleene algebras generalizes the one of regular expressions, and can be used to reason about conditional, loop, and simple (parametric) program manipulations.

Definition 1 (KAT). *A* Kleene algebra *is an idempotent semi-ring $\mathcal{K}$ endowed with a Kleene star operation $(-)^\star$, satisfying the following properties: for all $p, q, r \in \mathcal{K}$:*

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

where the ordering $\leq$ is defined as

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

A Kleene algebra with tests *(KAT, for short) is a pair* $(\mathcal{K}, \mathcal{B})$*, where* $\mathcal{K}$ *is a Kleene algebra of* actions *and* $\mathcal{B} \subseteq \mathcal{K}$ *is a boolean sub-algebra of* tests*. We call the class of all Kleene algebras with tests* **KATs***. We sometimes omit* $\mathcal{B}$ *if it can be inferred from the context. Tests are ranged over by* a, b, c, d*, whereas* actions *are ranged over by* p, q, r, s*.*

KATs can model program behavior by using actions to represent basic components, tests to represent guards, multiplication to represent sequential composition, addition to represent random choice, and star to represent iteration. Concretely, Kozen [1997] showed that KATs can be used to model while programs using the following encoding:

$$\texttt{if } b \texttt{ then } p \texttt{ else } q \triangleq bp + \overline{b}q$$

$$\texttt{while } b \texttt{ do } p \triangleq (bp)^\star \overline{b}$$

In the following sections, we will use KAT formulas to reason about the validity of Hoare logic and incorrectness logic triples. Following Kozen [2000] we will see judgments in these logic as predicates which can be expressed as KAT equalities. To do this, we will first need to define KAT terms, their interpretation in a KAT, and what it means for a predicate to be expressible using KATs.

We first need a notion of *alphabet*, which is a pair (K, B) of two disjoint finite sets: an *action alphabet* K and a *test alphabet* B. We will refer to the elements of K as *primitive actions*, ranged over by p, r, q, similarly to actions, and the elements of B as *primitive tests*, ranged over by a, b, c, similarly to tests.

We can now define the set of KAT terms.

Definition 2 (KAT Terms)**.** *The set* $\mathrm{KATTerm}_{K,B}$ *of* KAT terms *over the alphabet* (K, B) *is generated by the following grammar:*

$$\mathrm{t} \triangleq \mathrm{p} \in K \mid \mathrm{b} \in B \mid 0 \mid 1 \mid \mathrm{t}_1 + \mathrm{t}_2 \mid \mathrm{t}_1\mathrm{t}_2 \mid \mathrm{t}^\star \mid \overline{\mathrm{t_b}},$$

where t_b *does not contain primitive actions.*

Terms can be interpreted using a valuation of the primitive actions and tests in a KAT.

Definition 3 (KAT Valuation and Interpretation)**.** *Let us consider an alphabet* (K, B) *and a KAT* $(\mathcal{K}, \mathcal{B})$*. A* valuation *is a function* $u : K \cup B \to \mathcal{K}$ *such that* $u(\mathrm{b}) \in \mathcal{B}$ *for every* $\mathrm{b} \in B$*.*

Given a valuation u, we define the interpretation $[\![-]\!]_u : \mathrm{KATTerm}_{K,B} \to \mathcal{K}$ *as:*

$$\begin{aligned}
[\![\mathrm{p}]\!]_u &\triangleq u(\mathrm{p}) && \text{if } \mathrm{p} \in K \cup B \\
[\![\mathrm{t}_1 + \mathrm{t}_2]\!]_u &\triangleq [\![\mathrm{t}_1]\!]_u + [\![\mathrm{t}_2]\!]_u \\
[\![\mathrm{t}_1\mathrm{t}_2]\!]_u &\triangleq [\![\mathrm{t}_1]\!]_u [\![\mathrm{t}_2]\!]_u \\
[\![\mathrm{t}^\star]\!]_u &\triangleq [\![\mathrm{t}]\!]_u{}^\star \\
[\![\overline{\mathrm{t_b}}]\!]_u &\triangleq \overline{[\![\mathrm{t_b}]\!]_u} && \text{if } \mathrm{t_b} \text{ does not contain primitive actions}
\end{aligned}$$

Using the notion of interpretation, we can now define what it mean for an equality between KAT terms to be *valid*, which informally means that the equality holds for every valuation.

Definition 4 (Validity of KAT Equality)**.** *Given an alphabet* (K, B) *and two KAT terms* $\mathrm{t}_1, \mathrm{t}_2 \in \mathrm{KATTerm}_{K,B}$*, a statement* $\mathrm{t}_1 = \mathrm{t}_2$ *is* valid under the valuation $u : K \cup B \to \mathcal{K}$ *(denoted by* $\models_u \mathrm{t}_1 = \mathrm{t}_2$*), if*

$$[\![\mathrm{t}_1]\!]_u = [\![\mathrm{t}_2]\!]_u$$

A statement $\mathrm{t}_1 = \mathrm{t}_2$ *is* valid *in all KATs, denoted as*

$$\textbf{KATs} \models \mathrm{t}_1 = \mathrm{t}_2,$$

if $\mathrm{t}_1 = \mathrm{t}_2$ *is valid under all KAT valuations.*

We can now state formally how we can use the equational theory of KATs to reason about predicates.

Definition 5 (Expressiveness of a KAT)**.** *Suppose that we have an alphabet K, B, a KAT $\mathcal{K}$, an n-ary* predicate $P : \mathcal{K}^n \to \mathbb{B}$ *and n primitives* $\mathrm{p}_1, \ldots, \mathrm{p}_n \in K \cup B$*. We say two terms* $\mathrm{t}_1, \mathrm{t}_2 \in \mathrm{KATTerm}_{K,B}$ express *the predicate P in $\mathcal{K}$ over* $\mathrm{p}_1, \ldots, \mathrm{p}_n$*, if for all valuations $u : K \cup B \to \mathcal{K}$:*

$$\models_u \mathrm{t}_1 = \mathrm{t}_2 \iff P([\![\mathrm{p}_1]\!]_u, [\![\mathrm{p}_2]\!]_u, \ldots [\![\mathrm{p}_n]\!]_u)$$

Hoare logic and incorrectness logic treat programs as a relation between input and output memories. Accordingly, relation-based KATs will be fundamental to formulate and manipulate these logics.

Definition 6 (Relational KAT)**.** *A* relational KAT $(\mathcal{R}, \mathcal{B})$ *over a set X is a KAT where*

$$\mathcal{R} \subseteq \mathcal{P}(X \times X)$$

and tests $\mathcal{B} \subseteq \mathcal{K}$, are subsets of identity relation on X:

$$\mathcal{B} \subseteq \mathcal{P}(\{(x, x) \mid x \in X\})$$

such that

- *the addition operator $+$ is the union of relations*
- *the multiplication operator is the sequential composition of relations: for $p, q \in \mathcal{R}$,*

$$pq = p \fatsemi q = \{(x, z) \mid \exists y \in X, (x, y) \in p, (y, z) \in q\}$$

- *The additive identity 0 is the empty relation $\emptyset$*
- *The multiplicative identity 1 is the identity relation on X:*

$$\{(x, x) \mid x \in X\}$$

- *the star operator is the reflexive transitive closure: for $p \in \mathcal{R}$*

$$p^\star = \bigcup_{n \in \mathbb{N}} p^n$$

- *The complement of a test $b \in \mathcal{B}$ is:*

$$\overline{b} = 1 \setminus b$$

Some of the previous definitions can be extended to the relational setting:

- *a* relational valuation *is a valuation in a relational KAT.*
- *a* relational interpretation *is an interpretation generated by a relational valuation*
- *a statement* $\mathrm{t}_1 = \mathrm{t}_2$ *is* relationally valid *(denoted* $\mathbf{RELs} \models \mathrm{t}_1 = \mathrm{t}_2$*), if it is valid with for all relational valuations.*
- *A predicate is* expressible in **RELs** *if there exists a pair of KAT terms that express the predicate in all relational KATs.*

One of the most important result in [Kozen and Smith 1997] is the completeness of KAT over relational KAT:

$$\mathbf{RELs} \models \mathrm{t}_1 = \mathrm{t}_2 \iff \mathbf{KATs} \models \mathrm{t}_1 = \mathrm{t}_2$$

This means all equalities that are valid in all relational KAT can be deduced using just the theory of KAT. This result relies crucially on the construction of so-called *language KATs*, whose carrier sets are guarded terms of actions. Kozen and Smith [1997] showed that every KAT term can be interpreted in such KATs [Kozen and Smith 1997, Section 3] and, following from [Pratt 1980], proved the completeness of relational KAT by the existence of an injective homomorphism from

$$\textsc{Composition}\ \frac{\{a\}\ p\ \{b\} \qquad \{b\}\ q\ \{c\}}{\{a\}\ p;q\ \{c\}} \qquad \textsc{Conditional}\ \frac{\{b \wedge c\}\ p\ \{d\} \qquad \{\neg b \wedge c\}\ q\ \{d\}}{\{c\}\ \texttt{if}\ b\ \texttt{then}\ p\ \texttt{else}\ q\ \{d\}}$$

$$\textsc{While}\ \frac{\{b \wedge c\}\ p\ \{c\}}{\{c\}\ \texttt{while}\ b\ \texttt{do}\ p\ \{\neg b \wedge c\}} \qquad \textsc{Consequence}\ \frac{b' \to b \qquad \{b\}\ p\ \{c\} \qquad c \to c'}{\{b'\}\ p\ \{c'\}}$$

Fig. 3. Propositional Hoare logic

any language KAT to a relational KAT and the completeness of language KATs. We will use similar techniques to obtain the completeness results in Section 4.

2.2 Hoare Logic

Hoare logic is a fundamental tool for specifying and proving the correctness of while-like programs. Following Kozen [2000], we consider here *propositional Hoare logic*, which involves partial correctness Hoare triple $\{b\}\ p\ \{c\}$ consisting of atomic propositions, propositional connectives and while-like programs. As usual, a Hoare triple $\{b\}\ p\ \{c\}$ means that if the program p terminates when run on a memory satisfying b, it will result in a memory satisfying c. Figure 3 shows the rules of propositional Hoare logic, which differ from the classical setting in their omission of the assignment rule.

In its essence, Hoare logic is an *over-approximation* logic. To see this, it is convenient to think about a program p as a relation between input memories and output memories, and to think about predicates b and c as sets of states. Given a program p and a predicate b, we can write $\mathrm{post}(p)(b)$ for the set of post-states, that is

$$\mathrm{post}(p)(b) = \{x \mid \exists y \in b, (y, x) \in p\}.$$

A partial-correctness Hoare triple $\{b\}\ p\ \{c\}$ is valid iff

$$\mathrm{post}(p)(b) \subseteq c.$$

In words, c over-approximates the set of memories which can be obtained from b by running the program p. This condition can be expressed by means of the codomain of a relation: if we set

$$\mathrm{cod}(r) \triangleq \{y \mid \exists x \in X, (x, y) \in r\},$$

then, for all a relational KAT $(\mathcal{R}, \mathcal{B})$, $b, c \in \mathcal{B}$ and $p \in \mathcal{R}$,

$$\{b\}\ p\ \{c\} \triangleq \mathrm{cod}(bp) \subseteq \mathrm{cod}(c).$$

Kozen [2000] showed that we can reason about the partial correctness of propositional Hoare logic in KAT. To do this we can use tests to represent pre and post-conditions, thus encode a partial correctness propositional Hoare triple $\{b\}\ p\ \{c\}$ as the KAT equality:

$$bp\overline{c} = 0$$

or equivalently

$$bp = bpc.$$

Indeed, we can show that in all relational KATs $(\mathcal{R}, \mathcal{B})$ and $p \in \mathcal{R}, b, c \in \mathcal{B}$,

$$\{b\}\ p\ \{c\} \iff bp = bpc \iff bp\overline{c} = 0.$$

$$\textsc{Empty}\ \frac{}{[b]\ p\ [\epsilon : \bot]} \qquad \textsc{Consequence}\ \frac{b \to b' \qquad [b]\ p\ [\epsilon : c] \qquad c' \to c}{[b']\ p\ [\epsilon : c']}$$

$$\textsc{Disjunction}\ \frac{[b_1]\ p\ [\epsilon : c_1] \qquad [b_2]\ p\ [\epsilon : c_2]}{[b_1 \vee b_2]\ p\ [\epsilon : (c_1 \vee c_2)]} \qquad \textsc{Identity}\ \frac{}{[b]\ \texttt{skip}\ [ok : b, er : 0]} \qquad \textsc{Composition-Fail}\ \frac{[a]\ p\ [er : b]}{[a]\ p; q\ [er : b]}$$

$$\textsc{Composition-Normal}\ \frac{[a]\ p\ [ok : b] \qquad [b]\ p\ [\epsilon : c]}{[a]\ p; q\ [\epsilon : c]} \qquad \textsc{Choice-Left}\ \frac{[b]\ p\ [\epsilon : c]}{[b]\ p + q\ [\epsilon : c]} \qquad \textsc{Choice-Right}\ \frac{[b]\ q\ [\epsilon : c]}{[b]\ p + q\ [\epsilon : c]}$$

$$\textsc{Assume}\ \frac{}{[a]\ \texttt{assume}\ b\ [ok : a \wedge b, er : 0]} \qquad \textsc{Error}\ \frac{}{[b]\ \texttt{Fail}\ [er : b]} \qquad \textsc{Iter-Zero}\ \frac{}{[b]\ p^\star\ [ok : b]}$$

$$\textsc{Iter-NonZero}\ \frac{[b]\ p^\star; p\ [\epsilon : c]}{[b]\ p^\star\ [\epsilon : c]} \qquad \textsc{Iter-Dependent}\ \frac{\forall n \in \mathbb{N},\ [b(n)]\ p\ [ok : b(n+1)]}{[b(0)]\ p^\star\ [ok : \exists n, b(n)]}$$

Fig. 4. Generic incorrectness logic proof rules [O'Hearn 2020]

Given that KAT is complete over relational KAT, we can determine the relational validity of a propositional Hoare triple just by the equational theory of KAT.

2.3 Incorrectness Logic

O'Hearn [2020] proposed incorrectness logic to reason about incorrect programs. This logic is also related to earlier works by de Vries and Koutavas [2011]. Like Hoare logic, incorrectness logic is built on triples of the form $[b]\ p\ [c]$, which denote *incorrectness specifications*, where c is a set of undesirable final states and b is a precondition. Intuitively, such a triple says that every memory in c needs to be reachable from b by executing p.

If Hoare logic is an "over-approximation" logic, incorrectness logic is an "under-approximation" logic: an incorrectness triple $[b]\ p\ [c]$ is valid if and only if

$$\mathrm{post}(p)(b) \supseteq c.$$

In other words, the post-condition c "under-approximates" the canonical post condition of b after executing p. When we only look at program that terminates normally, the previous definition can be expressed in relational KAT as

$$\mathrm{cod}(bp) \supseteq \mathrm{cod}(c).$$

As mentioned by O'Hearn [2020, Section 2], the definitions of incorrectness and Hoare triples are highly symmetric:

$$\{b\}\ p\ \{c\} \triangleq \mathrm{cod}(bp) \subseteq \mathrm{cod}(c)$$
$$[b]\ p\ [c] \triangleq \mathrm{cod}(bp) \supseteq \mathrm{cod}(c)$$

Since incorrect programs often lead to explicit errors, O'Hearn [2020] also considered incorrectness triples $[b]\ p\ [er : c]$, which mean that, in addition to satisfying c, we require that the

final states of p lead to an error. More generally, we use the notation $[b]\ p\ [\epsilon : c]$, where the *error code* $\epsilon \in \{ok, er\}$ signals whether the program terminated normally or not; hence, the unqualified notation $[b]\ p\ [c]$ is simply a shorthand for $[b]\ p\ [ok : c]$, when abnormal termination is not a concern. Informally, such general triples mean that

$$[b]\ p\ [\epsilon : c] \triangleq \mathrm{cod}(bp) \supseteq (\epsilon : c).$$

In Section 5, we will give a more concrete definition of this semantics in an extension of KAT.

The generic proof rules of incorrectness logic are listed in Figure 4. Following O'Hearn [2020], we formulate incorrectness logic for a language of commands which is essentially the same of KAT terms. Conditionals and loops can be encoded with an encoding that is similar to the one given by Kozen [1997].

$$\texttt{if}\ b\ \texttt{then}\ p\ \texttt{else}\ q \triangleq (\texttt{assume}\ b; p) + (\texttt{assume}\ \overline{b}; q)$$

$$\texttt{while}\ b\ \texttt{do}\ p \triangleq (\texttt{assume}\ b; p)^\star; \texttt{assume}\ \overline{b}$$

Following O'Hearn [2020] we also use $[b]\ p\ [ok : c_1, er : c_2]$ as a shorthand for two different rules. For example, the rule

$$\textsc{Unit}\ \frac{}{[a]\ 1\ [ok : a, er : 0]}$$

stands for the two rules

$$\textsc{Unit-Ok}\ \frac{}{[a]\ 1\ [ok : a]} \qquad \textsc{Unit-Er}\ \frac{}{[a]\ 1\ [er : 0]}.$$

The fragment of incorrectness logic we consider here is *propositional* in the sense of Kozen [2000]. In particular, this presentation omits rules for variables and mutation [O'Hearn 2020].

3 FORMULATING INCORRECTNESS LOGIC

We might hope that the symmetry between Hoare logic and incorrectness logic would help us express incorrectness triples by adapting Kozen's formulation [Kozen 2000]. However, it is not obvious how we can exploit this symmetry, since it involves the codomain operation, which does not appear in Kozen's formulation. This difficulty, unfortunately, is fundamental: KAT cannot express incorrectness logic.

Definition 7. *Given a relational KAT $(\mathcal{R}, \mathcal{B})$, $p \in \mathcal{R}$ and $b, c \in \mathcal{B}$, an Incorrectness Triple $[b]\ p\ [c]$ is* valid *with respect to $(\mathcal{R}, \mathcal{B})$, denoted $(\mathcal{R}, \mathcal{B}) \models [b]\ p\ [c]$, if*

$$\mathrm{cod}(bp) \supseteq \mathrm{cod}(c)$$

we consider the predicate of incorrectness triple: $inc(b, p, c) \triangleq [b]\ p\ [c]$. We write inc over primitive tests b, c *and primitive action* p *as* [b] p [c].

To show that incorrectness triple cannot be formulated using equality of KAT terms, we only need to show that [b] p [c] cannot be expressed in **RELs**. More explicitly, we need to show that for all K, B where $\mathsf{p} \in K$ and $\mathsf{b}, \mathsf{c} \in B$, there does *not* exist a pair of terms $\mathsf{t}_1, \mathsf{t}_2 \in \mathrm{KATTerm}_{K,B}$ s.t. for all relational valuations u:

$$\models_u \mathsf{t}_1 = \mathsf{t}_2 \iff \models_u [\mathsf{b}]\ \mathsf{p}\ [\mathsf{c}]$$

Theorem 1. [b] p [c] *cannot be expressed in* **RELs**.

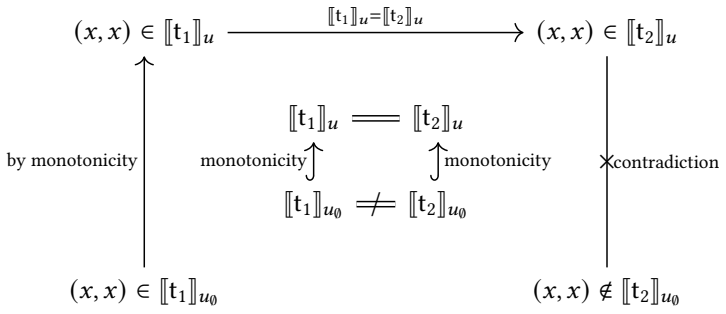

Fig. 5. Relation of interpretations u and $u_\emptyset$

Proof. Assume there is $t_1 = t_2$ that expresses [b] p [c]. Aiming to derive a contradiction, consider the following pair of valuations on b, c, p:

$$u_\emptyset(p) \triangleq \emptyset \qquad u(p) \triangleq \{(0,1)\}$$
$$u_\emptyset(b) \triangleq \{(0,0)\} \qquad u(b) \triangleq \{(0,0)\}$$
$$u_\emptyset(c) \triangleq \{(1,1)\} \qquad u(c) \triangleq \{(1,1)\}.$$

The only difference between u and $u_\emptyset$ is that p is mapped to $\emptyset$ in $u_\emptyset$. Note that the incorrectness triple [b] p [c] is valid for u, but not for $u_\emptyset$.

If we interpret a KAT term t as a nondeterministic program, the only way it can modify its state is if some of its action variables do (indeed, all other KAT operations keep the state intact). Thus, since $u_\emptyset(p) = \emptyset$, the elements of $[\![t_1]\!]_{u_\emptyset}$ and $[\![t_2]\!]_{u_\emptyset}$ must be either $(0,0)$ or $(1,1)$. Because the incorrectness triple [b] p [c] is invalid with $u_\emptyset$,

$$[\![t_1]\!]_{u_\emptyset} \neq [\![t_2]\!]_{u_\emptyset}.$$

Without loss of generality, assume that there exists $(x,x) \in [\![t_1]\!]_{u_\emptyset}$, but not in $[\![t_2]\!]_{u_\emptyset}$. We can show by induction that the interpretation of a KAT term is monotonic with respect to its action variables (this does not hold for test variables, since test terms can be negated). Since $u_\emptyset(p) \subseteq u(p)$, we have $(x,x) \in [\![t_1]\!]_u$. Moreover, we know that [b] p [c] is valid for u; thus $[\![t_2]\!]_u = [\![t_1]\!]_u$ and $(x,x) \in [\![t_2]\!]_u$. As depicted in Figure 5, we will conclude by proving that $(x,x) \in [\![t_2]\!]_{u_\emptyset}$, which is the opposite of what we assumed earlier, thus entailing a contradiction.

To demonstrate the contradiction, we need the following core lemma, proved by induction on t_2: if $(x,x) \in [\![t_2]\!]_u$, then either (1) $(x,x) \in [\![t_2]\!]_{u_\emptyset}$, or (2) there exists z and z', s.t. $(x,z) \in u(p)$ and $(z',x) \in u(p)$. Intuitively, if the element (x,x) is generated by purely by some tests in $[\![t_2]\!]_u$, then we can ignore all the action variables in the term by setting it to $\emptyset$. Otherwise, (x,x) must be generated by composing some actions together, since tests only *filters out* elements when composed and cannot generated new elements. Thus we will need at least a action to start with x, and another action to end with x.

However, the second case cannot hold. Indeed, since $u(p)$ is a singleton, it is equivalent to saying that $(x,x) \in u(p) = \{(0,1)\}$, which is impossible. Thus, we learn that $(x,x) \in [\![t_2]\!]_{u_\emptyset}$, yielding the sought contradiction. We conclude that there can't be an equation $t_1 = t_2$ that expresses incorrectness triples.

□

One strategy for exploiting the symmetry between Hoare and incorrectness logic is to extend KAT with a codomain operator. Similar ideas have been explored in prior work [Desharnais et al. 2004; Fahrenberg et al. 2021; Gomes and Struth 2016]. However, rather than adopting a full-fledged codomain operator, it suffices for our purposes to consider a equational theory that only extends KAT with a top element. Such an algebraic structure has also been considered in prior application [Mamouras 2017, Section 4], though for a different purpose.

Definition 8 (KAT with a Top Element)**.** *A KAT with top, or TopKAT, is a KAT $\mathcal{K}$ that contains a largest element $\top$; that is, for all elements $p \in \mathcal{K}$,*

$$\top \geq p.$$

A relational TopKAT over X is a relational KAT where the top element is the complete relation $X \times X$.

Theorem 2 (TopKAT can Express Codomain)**.** *For all relational TopKATs $\mathcal{R}$, and $p, q \in \mathcal{R}$ The following is true:*

$$\top p = \top q \iff \mathrm{cod}(p) = \mathrm{cod}(q)$$

and

$$\top p \leq \top q \iff \mathrm{cod}(p) \subseteq \mathrm{cod}(q).$$

Proof. For all relations $r \subseteq X \times X$:

$$\top \fatsemi r = \{(z, y) \mid z \in X \wedge \exists x, (x, y) \in r\} = \{(z, y) \mid z \in X \wedge y \in \mathrm{cod}(r)\}$$

Therefore for two relation p, q, we will have

$$\begin{aligned} \top \fatsemi q = \top \fatsemi p &\iff \{(z, y) \mid z \in X \wedge y \in \mathrm{cod}(q)\} = \{(z, y) \mid z \in X \wedge y \in \mathrm{cod}(p)\} \\ &\iff \mathrm{cod}(q) = \mathrm{cod}(p) \\ \top \fatsemi q \leq \top \fatsemi p &\iff \{(z, y) \mid z \in X \wedge y \in \mathrm{cod}(q)\} \subseteq \{(z, y) \mid z \in X \wedge y \in \mathrm{cod}(p)\} \\ &\iff \mathrm{cod}(q) \subseteq \mathrm{cod}(p) \end{aligned}$$

□

Corollary 1. *For all relational TopKATs $(\mathcal{R}, \mathcal{B})$, and $p \in \mathcal{R}$, $b, c \in \mathcal{B}$ we have the following:*

$$\top bp \geq \top c \iff [b]\ p\ [c]$$

Notice that the left hand side of this equivalence makes sense in all TopKATs, not just relational ones. Thus it is natural to generalize the definition of incorrectness triple to all TopKATs using the inequality $\top bp \geq \top c$. As a matter of fact, just by using the equational theory of TopKAT, we can find several equivalent formulations of incorrectness triple:

Theorem 3 (Equivalent Formulation of Incorrectness Logic)**.** *Given a TopKAT $(\mathcal{K}, \mathcal{B})$, where $p \in \mathcal{K}$ and $b, c \in \mathcal{B}$, we say that an incorrectness triple $[b]\ p\ [c]$ is valid if the following equivalent conditions are met:*

$$\top bp \geq \top c \iff \top bp \geq c \iff \top bpc = \top c$$

A similar encoding involving $\top$ was also mentioned by O'Hearn [2020, Section 3.5].

We want to show that this abstraction is enough to capture all the rules of incorrectness logic. Indeed most of the the rules are formulated using program operations expressible in **KATs** [O'Hearn 2020]. We focus here on the proof rules concerning normal program termination, and we will further investigate the rules with error in Section 5.

In Figure 6, we present all the rules of incorrectness logic with normal terminations using the syntax of KAT. (Note that this differs slightly from O'Hearn's original notation; for instance, the

$$\textsc{Empty}\ \frac{}{[b]\ p\ [0]} \qquad \textsc{Consequence}\ \frac{b \le b' \qquad [b]\ p\ [c] \qquad c' \le c}{[b']\ p\ [c']} \qquad \textsc{Disjunction}\ \frac{[b_1]\ p\ [c_1] \qquad [b_2]\ p\ [c_2]}{[b_1 + b_2]\ p\ [c_1 + c_2]}$$

$$\textsc{Identity}\ \frac{}{[b]\ 1\ [b]} \qquad \textsc{Composition}\ \frac{[a]\ p\ [b] \qquad [b]\ q\ [c]}{[a]\ pq\ [c]} \qquad \textsc{Choice-Left}\ \frac{[a]\ p\ [b]}{[a]\ p + q\ [b]} \qquad \textsc{Choice-Right}\ \frac{[a]\ q\ [b]}{[a]\ p + q\ [b]}$$

$$\textsc{Assume}\ \frac{}{[b]\ c\ [bc]} \qquad \textsc{Iter-Zero}\ \frac{}{[b]\ p^\star\ [b]} \qquad \textsc{Iter-NonZero}\ \frac{[b]\ p^\star p\ [c]}{[b]\ p^\star\ [c]} \qquad \textsc{Iter-Dependent}\ \frac{\forall n \in \mathbb{N}, [b_n]\ p\ [b_{n+1}]}{[b_0]\ p^\star\ [\sup_{n\in\mathbb{N}} b_n]}$$

Fig. 6. Incorrectness logic with normal termination

sequencing operator $p; q$ corresponds to multiplication pq in KAT, and the logical implication $b \to b'$ corresponds to order $b \le b'$.) For the Iter-dependent rule (Backwards Variant in [O'Hearn 2020]), the function $b(n)$ corresponds to a sequence of tests $(b_n)_{n\in\mathbb{N}}$, and the existential $\exists n, b(n)$ corresponds to the infinite disjunction of all these b_n, which we express as $\sup_{n\in\mathbb{N}} b_n$. (Note that sup does not exist in KATs, TopKATs, or boolean algebras in general, and this rule implicitly assumes that $\sup_{n\in\mathbb{N}} b(n)$ exists.)

Theorem 4 (Soundness of *ok* State Rules)**.** *All the rules of Figure 6 are derivable in all TopKATs.*

Proof. We give the proof of composition rule as an example. By unfolding the definition of incorrectness triple, we have to show the following inequality:

$$\top a(pq) \ge c$$

first, by idempotency of $\top$ and associativity of multiplication, we have

$$\top a(pq) = \top\top apq$$

By order preserving of multiplication, and $\top ap \ge b$, we have

$$\top a(pq) = \top\top apq \ge \top bq \ge c$$

□

An alternative encoding for Hoare logic. Since **TopKATs** can express codomain (Theorem 2), we can exploit the symmetry between incorrectness logic and Hoare logic (Section 2.2) to give another encoding of Hoare logic in **TopKATs**:

$$\top bp \le \top c.$$

This is equivalent to the original encoding $bp = bpc$ proposed for KATs [Kozen 2000]. Since the proof rules of propositional Hoare logic are sound for that encoding [Kozen 2000, Theorem 3.1], they are also sound for ours.

Theorem 5 (Equivalence of Hoare Logic Formulations)**.** *For all TopKATs $(\mathcal{K}, \mathcal{B})$, and three element $p \in \mathcal{K}$ and $b, c \in \mathcal{B}$, the following inequalities are equivalent:*

$$bp \le \top c \iff \top bp \le \top c \iff bp = bpc.$$

Proof. • We first show the equivalence:

$$bp \leq \top c \iff \top bp \leq \top c.$$

$\Longleftarrow$ is by $1 \leq \top$, thus

$$bp \leq \top bp \leq \top c,$$

and $\Longrightarrow$ is by the idempotency of $\top$, multiply both side with $\top$, we have

$$\top bp \leq \top\top c = \top c.$$

• Then we show the equivalence

$$bp \leq \top c \iff \top bp = bpc.$$

Start with $\Longleftarrow$, because $\top \geq bp$, therefore we have

$$bp = bpc \leq \top c.$$

Then $\Longrightarrow$, we start from $bp \leq \top c$, we first multiply $\overline{c}$ on both side,

$$bp\overline{c} = \top c\overline{c} = \top 0 = 0,$$

then we add bpc to both side, we have

$$bp = bp(\overline{c} + c) = 0 + bpc = bpc.$$

□

4 PROPERTIES OF TOPKATS

In the previous section, we have shown that the theory of TopKAT subsumes incorrectness logic. In this section, we study some of the properties of its algebraic equational theory. First, we will show that TopKAT is *incomplete* with respect to relational TopKATs: there are identities that are valid for every relational TopKAT that cannot be proven using the TopKAT identities alone. This contrasts with what happens for KAT, which is complete for relational KATs. We will then show the completeness of TopKATs with respect to a class of *language-based* TopKATs and also with respect to a more general class of *relational* TopKATs, as well as the decidability of TopKAT equalities. Finally we will introduce the concept of equational expressiveness, and show that the general relational TopKAT has the same expressiveness as relational KAT, hence cannot express incorrectness logic.

4.1 Incompleteness with Respect to Relational TopKATs

We can naturally extend the definition of term, and primitives of a term from KAT to TopKAT. An *alphabet* (K, B) is two disjoint sets K and B, where the elements of K are called *primitive actions*, and the elements of B are called *primitive tests*. The set $\mathrm{TopKATTerm}_{K,B}$ of *TopKAT terms* over an alphabet K, B is generated by the following grammar:

$$\mathrm{t} \triangleq \mathrm{p} \in K \mid \mathrm{b} \in B \mid \top \mid 0 \mid 1 \mid \mathrm{t}_1 + \mathrm{t}_2 \mid \mathrm{t}_1\mathrm{t}_2 \mid \mathrm{t}^\star \mid \overline{\mathrm{t_b}},$$

where $\mathrm{t_b}$ does not contain primitive actions.

Similarly we can extend the notions of valuation and interpretation. Given an alphabet K, B, and a TopKAT $\mathcal{K}$, a valuation for TopKAT terms is a function $u : K \cup B \to \mathcal{K}$. The interpretation $[\![-]\!]_u : \mathrm{TopKATTerm}_{K,B} \to \mathcal{K}$ is generated by u, as in Definition 3.

We start with our negative result, which is the incompleteness over relational TopKATs.

Theorem 6. **TopKATs** *is incomplete over* **TopRELs***: the formulas*

$$(\top \mathsf{p})(\top \mathsf{p}) = \top \mathsf{p} \text{ and } \mathsf{p}\top\mathsf{p} \geq \mathsf{p}$$

are valid in every relational TopKAT, but not in every TopKAT.

Proof. We can show that the $(\top \mathsf{p})(\top \mathsf{p}) = \top \mathsf{p}$ holds in every relational TopKAT simply by unfolding the definitions. To exhibit a TopKAT where it does not hold, consider the following counterexample. We take a relational KAT over $\{0, 1\}$ whose largest element is

$$\top \triangleq \{(0, 0), (1, 1), (0, 1)\}.$$

(Note that this KAT is *not* a relational TopKAT: the largest element is not the complete relation.) Then, by taking a valuation $u(\mathsf{p}) \triangleq \{(0, 1)\}$, we see that

$$[\![\top \mathsf{p}]\!]_u = \{(0, 1)\}$$
$$[\![(\top \mathsf{p})(\top \mathsf{p})]\!]_u = \{(0, 1)\} \fatsemi \{(0, 1)\} = \emptyset \neq [\![\top \mathsf{p}]\!]_u$$

And the same counter example also works for $\mathsf{p}\top\mathsf{p} \geq \mathsf{p}$. □

The incompleteness result might be discouraging, since we might not be able to derive some valid theorems of incorrectness logic using TopKAT. However in Theorem 4 and Theorem 13, we have showed that all the rules of incorrectness logic are derivable using either the equational theory of TopKAT or FailTopKAT. Thus, our framework is at least as strong as the structural proof rules presented by O'Hearn [2020, Figure 2].

4.2 Completeness of TopKAT*s

Language models are essential in various extensions of Kleene algebra, since they are the basis of many completeness and decidability proofs [Brunet and Pous 2014; Kozen and Smith 1997]. In this section, we adapt this idea to the setting of TopKATs. The construction follows Kozen and Smith [1997, Section 3], except that both the carrier and the multiplicative identity will contain the term $\top$, and the coalesced product is also modified accordingly.

We augment the theory of TopKAT with the $\star$-continuity axiom, which will be helpful in reason about multiplication in later lemmas (see Kozen and Smith [1997, Lemma 4] for a demonstration of this technique). Later, we will use the completeness of TopKAT*to obtain the completeness result of TopKAT in Theorem 9.

Definition 9. *A $\star$-continuous TopKAT, or a TopKAT*, is a TopKAT that satisfies*

$$pq^\star r = \sup_{n \in \mathbb{N}} pq^n r,$$

where q^n is defined inductively:

$$q^0 = 1$$
$$q^{k+1} = q \cdot q^k.$$

We call the set of all TopKAT*s **TopKAT*s**.

Definition 10. *For an alphabet K, B, where $B = \{\mathsf{b}_0, \mathsf{b}_1, \ldots \mathsf{b}_n\}$ a* minimal test *(a smallest non-zero test in the free TopKAT, which is called "atom" by Kozen and Smith [1997]), is a term of the following form:*

$$\hat{\mathsf{b}_0}\hat{\mathsf{b}_1} \ldots \hat{\mathsf{b}_n} \text{ where } \forall i \in \{0, 1, \ldots, n\}, \hat{\mathsf{b}_i} \in \{\mathsf{b}_i, \overline{\mathsf{b}_i}\}$$

We let α, β, γ range over minimal tests. We will write $1_{\mathcal{G}}$ as the set of all minimal tests over an alphabet K, B, when K, B can be inferred from context. And we will also define $1_{\mathcal{G}_\top} \triangleq 1_{\mathcal{G}} \cup \top$

Definition 11. *For an alphabet K, B a* guarded term *is a term following form:*

$$\alpha_0 \mathrm{p}_1 \alpha_1 \dots \mathrm{p}_n \alpha_n$$

where $\mathrm{p}_1, \mathrm{p}_2, \dots \mathrm{p}_n \in K$ and $\alpha_1, \alpha_2, \dots \alpha_n \in 1_{\mathcal{G}}$.

We sometimes write a guarded term as $\mathrm{s}\alpha$ or $\alpha\mathrm{s}$, to represent the concatenation of the term s with a minimal test α. We write $GT_\top$ as the set of all guarded terms with the term $\top$.

Definition 12. *The* coalesced product *is a partial binary operation on $GT_\top$ defined as follows:*

$$\mathrm{s}_1 \diamond \mathrm{s}_2 = \begin{cases} \hat{\mathrm{s}_1}\alpha\hat{\mathrm{s}_2} & \mathrm{s}_1 = \hat{\mathrm{s}_1}\alpha \text{ and } \mathrm{s}_2 = \alpha\hat{\mathrm{s}_2} \\ \top & \mathrm{s}_1 = \mathrm{s}_2 = \top \\ \textit{undefined} & \textit{otherwise} \end{cases}$$

This definition can naturally extend to subsets of $GT_\top$ as a total binary operation on subsets of $GT_\top$:

$$S_1 \diamond S_2 = \{\mathrm{s}_1 \diamond \mathrm{s}_2 \mid \mathrm{s}_1 \in S_1, \mathrm{s}_2 \in S_2, \mathrm{s}_1 \diamond \mathrm{s}_2 \textit{ is defined}\}$$

It is not hard to show that the coalesced product is associative, distributes over infinite and finite unions, and has an identity $1_{\mathcal{G}_\top}$. Thus the set of all guarded terms with top forms a TopKAT*:

Definition 13. *Given an alphabet K, B, the* language TopKAT* *or $\mathcal{G}_\top$ is a TopKAT* where*

- *The elements are subsets of $GT_\top$*
- *The tests are subsets of $1_{\mathcal{G}_\top}$*
- *The multiplication is coalesced product*
- *The addition is set union*
- *The star operator is defined as follows:*

$$S^\star = \bigcup_{n \in \mathbb{N}} S^n \textit{ where } S^0 = 1_{\mathcal{G}_\top} \textit{ and } S^{k+1} = S \diamond S^k$$

- *The complement of test b is $1_{\mathcal{G}_\top} - b$*
- *the top element is the set $GT_\top$*
- *the additive identity is the empty set*
- *the multiplicative identity is $1_{\mathcal{G}_\top}$*

Definition 14. *Given an alphabet K, B, the* standard valuation *is $g : K \cup B \to \mathcal{G}_\top$ defined as follows:*

$$g(\mathrm{p}) = \{\alpha \mathrm{p} \beta \mid \alpha\beta \in 1_{\mathcal{G}}\}$$
$$g(\mathrm{b}) = \{\alpha \mid \mathrm{b} \textit{ appear positively in } \alpha\}$$

We call the interpretation $[\![-]\!]_g : \mathrm{TopKATTerm}_{K,B} \to \mathcal{G}_\top$ the standard interpretation.

With all the necessary definitions in place, we can recreate the completeness lemma [Kozen and Smith 1997, Lemma 4]. The proof of the lemma is to simply extend Kozen and Smith's proof with a $\top$ case.

Lemma 1. *For all TopKAT*s $\mathcal{K}$, all valuations $u : K \cup B \to \mathcal{K}$ and three terms $\mathrm{t}_1, \mathrm{t}_2, \mathrm{t}_3$:*

$$[\![\mathrm{t}_1\ \mathrm{t}_2\ \mathrm{t}_3]\!]_u = \sup_{\mathrm{s} \in [\![t_2]\!]_g} [\![\mathrm{t}_1\ \mathrm{s}\ \mathrm{t}_3]\!]_u$$

where the sup *is respect to the ordering in $\mathcal{K}$. A special case of this result is to let $\mathrm{t}_1 = \mathrm{t}_3 = 1$, then*

$$[\![\mathrm{t}_2]\!]_u = \sup_{\mathrm{s} \in [\![t_2]\!]_g} [\![\mathrm{s}]\!]_u$$

Theorem 7 (completeness over standard interpretation)**.** *For all alphabets K, B, and terms $\mathrm{t}_1, \mathrm{t}_2 \in \mathrm{TopKATTerm}_{K,B}$:*

$$\mathbf{TopKAT}^\star\mathbf{s} \models \mathrm{t}_1 = \mathrm{t}_2 \iff \models_g \mathrm{t}_1 = \mathrm{t}_2$$

Proof. $\Longrightarrow$, because g is a valuation to a TopKAT*, then if $\mathrm{t}_1 = \mathrm{t}_2$ is valid under all valuations to TopKAT*, then it is valid under g.

$\Longleftarrow$, by Lemma 1 for all TopKAT*s $\mathcal{K}$ and valuations $u : K \cup B \to \mathcal{K}$,

$$[\![\mathrm{t}_1]\!]_u = \sup_{\mathrm{s} \in [\![t_1]\!]_g} [\![\mathrm{s}]\!]_u \text{ and } [\![\mathrm{t}_2]\!]_u = \sup_{\mathrm{s} \in [\![t_2]\!]_g} [\![\mathrm{s}]\!]_u.$$

Therefore because $[\![t_1]\!]_g = [\![t_2]\!]_g$, for all valuations u,

$$[\![\mathrm{t}_1]\!]_u = \sup_{\mathrm{s} \in [\![t_1]\!]_g} [\![\mathrm{s}]\!]_u = \sup_{\mathrm{s} \in [\![t_2]\!]_g} [\![\mathrm{s}]\!]_u = [\![\mathrm{t}_2]\!]_u.$$

Therefore $\mathrm{t}_1 = \mathrm{t}_2$ is valid in all valuations, thus

$$\mathbf{TopKAT}^\star\mathbf{s} \models \mathrm{t}_1 = \mathrm{t}_2.$$

□

Besides language TopKATs and relational TopKATs, we have already mentioned a more general class of relation-based TopKATs in the proof of Theorem 6, where the top element is not necessarily the complete relation. The motivation to investigate this class of TopKATs is more than simple mathematical curiosity: being a relation-based class of TopKATs, these structures has the potential to model programs as a input and output relations, just like relational TopKATs.

Definition 15. *A* general relational TopKAT *is a relational TopKAT where the top element is not necessarily the complete relation. We denote all the general relational TopKATs* **TopGRELs**.

Because composition distributes over infinite unions of relations, we can show the $\star$-continuity axiom holds in **TopGRELs**. Thus, all general relational TopKATs are TopKAT*s.

Example 1. *The following elements form a general relational TopKAT over $\{0, 1\}$, but not a relational TopKAT:*

$$\emptyset$$
$$\{(1, 1)\}$$
$$\{(0, 1)\}$$
$$\{(0, 0), (1, 1)\}$$
$$\{(0, 1), (0, 0), (1, 1)\}$$

where the top element is $\{(0, 1), (0, 0), (1, 1)\}$, not the complete relation on $\{0, 1\}$

We can extend some definitions about relational TopKATs to general relational TopKATs.

Definition 16. *Given an alphabet K, B*

- *for all general relational TopKATs $\mathcal{R}$, a* general relational valuation *is a function*
$$r : K \cup B \to \mathcal{R};$$
- *A* general relational interpretation
$$[\![-]\!]_r : \mathrm{TopKATTerm}_{K,B} \to \mathcal{R}$$
is the interpretation generated by r as in Definition 3;

- *For two term* $t_1, t_2 \in \mathrm{TopKATTerm}_{K,B}$*, the statement* $t_1 = t_2$ *is* valid under all general relational interpretations *if for all general relational valuations* r*:*

$$[\![t_1]\!]_r = [\![t_2]\!]_r,$$

we write it as

$$\mathbf{TopGRELs} \models t_1 = t_2.$$

Theorem 8 (Completeness over **TopGRELs**). *Given an alphabet* K, B*, and two TopKAT terms* $\mathrm{t}_1, \mathrm{t}_2 \in \mathrm{TopKATTerm}_{K,B}$*,*

$$\mathbf{TopGRELs} \models \mathrm{t}_1 = \mathrm{t}_2 \iff \mathbf{TopKAT^\star s} \models \mathrm{t}_1 = \mathrm{t}_2$$

Proof. Same proof as in [Kozen and Smith 1997, Lemma 5, Theorem 6], we define the following injective homomorphism from a language TopKAT to a general relational TopKAT

$$h(S) = \{(s_1, s_1 \diamond s) \mid s_1 \in GT_\top, s \in S\}$$

We first verify that it is homomorphism, most of the cases is the same as [Kozen and Smith 1997], the only new case is to show that $h(GT_\top)$ is the top element, which is obvious.

Proving $\Longleftarrow$: Because every general relation TopKAT is a TopKAT*. If a statement is true for all TopKAT*s, it is true for all general relational TopKATs.

Proving $\Longrightarrow$: since h is an injective homomorphism, the domain is isomorphic to its range. Thus for all language TopKATs, there exists a isomorphic general relational TopKAT. Therefore if $\mathrm{t}_1 = \mathrm{t}_2$ is true in all general relational TopKATs, it is true in all the ranges of h. Hence the $\mathrm{t}_1 = \mathrm{t}_2$ is also true in all language TopKATs. Finally by completeness over standard interpretation, we have **TopKAT*s** is complete over **TopGRELs**. □

4.3 Completeness and Decidability of TopKAT Equality

Though we already have completeness of **TopKAT*s**, we still need to show **TopKATs** is complete over the previous classes of TopKATs. Following the same strategy in [Kozen and Smith 1997, Section 7], we construct a regular expression that is "equivalent" to each TopKAT term. Since KA equivalence is subsumed by TopKAT equivalence and KA equivalence is known to be decidable, thus we can have completeness and decidability of TopKAT equivalence.

First, let's recall some definitions regarding regular expressions:

Definition 17. *The function* R *takes a regular expression with alphabet* K *and returns its regular set. where a regular expression is a term* t *defined as follows:*

$$\mathrm{t} \triangleq \mathrm{p} \in K \mid 1 \mid 0 \mid \mathrm{t}_1\mathrm{t}_2 \mid \mathrm{t}_1 + \mathrm{t}_2 \mid \mathrm{t}^\star$$

And R *is defined inductively as follows:*

$$\begin{aligned}
R(0) &\triangleq \emptyset \\
R(1) &\triangleq \{\epsilon\} \text{ where } \epsilon \text{ is the empty string} \\
R(\mathrm{p}) &\triangleq \{\mathrm{p}\} \\
R(\mathrm{t}_1 + \mathrm{t}_2) &\triangleq R(\mathrm{t}_1) \cup R(\mathrm{t}_2) \\
R(\mathrm{t}_1\mathrm{t}_2) &\triangleq \{s_1 s_2 \mid s_1 \in R(\mathrm{t}_1), s_2 \in R(\mathrm{t}_2)\} \text{ where } s_1 s_2 \text{ is the concatenation of } s_1 \text{ and } s_2 \\
R(\mathrm{t}^\star) &\triangleq \bigcup_{n \in \mathbb{N}} (R(\mathrm{t}))^n
\end{aligned}$$

We also note that given an alphabet K, B, all the terms in $\mathrm{TopKATTerm}_{K,B}$ are regular expressions over the alphabet

$$K \cup B \cup \overline{B} \cup \{\top\},$$

where $\overline{B} = \{\overline{\mathsf{b}} \mid \mathsf{b} \in B\}$. Thus the function R can be applied onto elements of $\mathrm{TopKATTerm}_{K,B}$.

Hence, following Kozen and Smith [1997], for each term $\mathsf{t} \in \mathrm{TopKATTerm}_{K,B}$, we can construct a term $r(\mathsf{t}) \in \mathrm{TopKATTerm}_{K,B}$ (similar to $\hat{\mathsf{t}}$ in [Kozen and Smith 1997]) that is equivalent to the original term in several ways.

Lemma 2. *Given an alphabet (K, B), there exists a function $r : \mathrm{TopKATTerm}_{K,B} \to \mathrm{TopKATTerm}_{K,B}$, s.t. for all terms $\mathsf{t} \in \mathrm{TopKATTerm}_{K,B}$*

- **TopKATs** $\models \mathsf{t} = r(\mathsf{t})$
- $R(r(\mathsf{t})) = [\![r(\mathsf{t})]\!]_g$

Note that this theorem is similar to [Kozen and Smith 1997, Lemma 7]. However, we did not include 7(iii), which is just a induction hypothesis to show that the construction is sound. Since it is not used later, we will omit it from the theorem.

Theorem 9 (Completeness of **TopKATs**). *Given an alphabet K, B, and two TopKAT terms $\mathsf{t}_1, \mathsf{t}_2 \in \mathrm{TopKATTerm}_{K,B}$ The following conditions are equivalent*

- **TopKATs** $\models \mathsf{t}_1 = \mathsf{t}_2$
- **TopGRELs** $\models \mathsf{t}_1 = \mathsf{t}_2$
- $\models_g \mathsf{t}_1 = \mathsf{t}_2$
- **KAs** $\models r(\mathsf{t}_1) = r(\mathsf{t}_1)$

Proof.

- First we show that

$$\models_g \mathsf{t}_1 = \mathsf{t}_2 \iff \textbf{KAs} \models r(\mathsf{t}_1) = r(\mathsf{t}_2)$$

$\Longleftarrow$, since all TopKATs are KAs, We have

$$\textbf{TopKATs} \models r(\mathsf{t}_1) = r(\mathsf{t}_2)$$

and by Lemma 2,

$$\textbf{TopKATs} \models r(\mathsf{t}_1) = \mathsf{t}_1 \text{ and } \textbf{TopKATs} \models r(\mathsf{t}_2) = \mathsf{t}_2$$

Therefore we have Therefore we have Therefore we have

$$\textbf{TopKATs} \models \mathsf{t}_1 = \mathsf{t}_2$$

and since g is a TopKAT valuation, we have

$$\models_g \mathsf{t}_1 = \mathsf{t}_2.$$

then $\Longrightarrow$, by Lemma 2

$$R(r(\mathsf{t}_1)) = [\![\mathsf{t}_1]\!]_g \text{ and } R(r(\mathsf{t}_2)) = [\![\mathsf{t}_2]\!]_g$$

Therefore by completeness over language model of KA [Kozen 1994, Theorem 19], we have **KAs** $\models \mathsf{t}_1 = \mathsf{t}_2$,

- Then we show

$$\models_g \mathsf{t}_1 = \mathsf{t}_2 \iff \textbf{TopKATs} \models \mathsf{t}_1 = \mathsf{t}_2$$

$\Longleftarrow$ is trivial, since g is a TopKAT interpretation. $\Longrightarrow$ is shown as follows: we have

$$\models_g \mathsf{t}_1 = \mathsf{t}_2 \iff \textbf{KAs} \models r(\mathsf{t}_1) = r(\mathsf{t}_1)$$

since every TopKAT is a KA, therefore

$$\textbf{TopKATs} \models r(\mathsf{t}_1) = r(\mathsf{t}_2).$$

Since **TopKATs** $\models r(\mathrm{t}_1) = \mathrm{t}_1$, **TopKATs** $\models r(\mathrm{t}_2) = \mathrm{t}_2$, we have **TopKATs** $\models \mathrm{t}_1 = \mathrm{t}_2$.

- Finally we show

$$\models_g \mathrm{t}_1 = \mathrm{t}_2 \iff \textbf{TopGRELs} \models \mathrm{t}_1 = \mathrm{t}_2$$

By completeness over standard interpretation over **TopKAT*s** (Lemma 1) and the completeness over general relational TopKATs over **TopKAT*s** (Theorem 8) We have

$$\models_g \mathrm{t}_1 = \mathrm{t}_2 \iff \textbf{TopKAT}^\star\textbf{s} \models \mathrm{t}_1 = \mathrm{t}_2 \iff \textbf{TopGRELs} \models \mathrm{t}_1 = \mathrm{t}_2$$

□

Corollary 2 (decidability of TopKAT equivalence). *For all alphabets K, B, and terms $\mathrm{t}_1, \mathrm{t}_2 \in \mathrm{TopKATTerm}_{K,B}$, the equality* **TopKATs** $\models \mathrm{t}_1 = \mathrm{t}_2$ *is decidable.*

Proof. by Theorem 9, we have

$$\textbf{KAs} \models r(\mathrm{t}_1) = r(\mathrm{t}_1) \iff \textbf{TopKATs} \models \mathrm{t}_1 = \mathrm{t}_2$$

If we constrain the alphabet to only include primitives that appeared in t_1 and t_2, then the size of $r(\mathrm{t}_1)$ and $r(\mathrm{t}_2)$ will be exponential of the size of t_1 and t_2. Since Kleene algebra is is decidable with PSPACE complexity, and $r(t_1), r(t_2)$ takes exponential space to store, we can first compute and store $r(t_1), r(t_2)$ and then decide their equality under **KAs**. This algorithm will have EXPSPACE complexity. □

4.4 PSPACE Algorithm

Cohen et al. [1999, Section 4.2] has shown that there is no way to avoid the exponential size blowup for $r(\mathrm{t})$ defined in Lemma 2. Thus there is a fundamental complexity barrier to the method for deciding a TopKAT equality by reducing to KA equalities. However, Cohen et al. developed a way to get around this barrier and showed deciding KAT equalities is PSPACE complete.

However, directly translate Cohen et al.'s insight from KAT to TopKAT can be problematic, as the addition of $\top$ can cause complications in the proof of the core lemmas. Thus we consider the alternative of reducing a TopKAT term into a KAT term, and deal with $\top$ separately from guarded terms.

Given a TopKAT term $\mathrm{t} \in \mathrm{TopKATTerm}_{K,B}$, we can reduce it into a KAT term $k(\mathrm{t}) \in \mathrm{KATTerm}_{K,B}$ where we just replace all the $\top$ in t with

$$(\sum_{\mathrm{p} \in K} \mathrm{p})^\star.$$

Then the size of $k(\mathrm{t})$ is polynomial to the the alphabet size plus the size of t.

Lemma 3. *For all guarded term $\alpha_0 \mathrm{p}_1 \alpha_1 \dots \mathrm{p}_m \alpha_m$*

$$\alpha_0 \mathrm{p}_1 \alpha_1 \dots \mathrm{p}_m \alpha_m \in [\![\mathrm{t}]\!]_g \iff \alpha_0 \mathrm{p}_1 \alpha_1 \dots \mathrm{p}_m \alpha_m \in G(\mathrm{k}(\mathrm{t}))$$

where G is the standard interpretation for KAT terms [Kozen and Smith 1997, Section 3.1].

Lemma 4. *For all term* t*, deciding whether $\top \in [\![\mathrm{t}]\!]_g$ takes polynomial time, hence in PSPACE.*

To decide whether two TopKAT term t and t′ are equal, we need to decide whether $[\![\mathrm{t}]\!]_g = [\![\mathrm{t}']\!]_g$. By Savitch's Theorem, we can non-deterministically guess a term in $GT_\top$, then decide whether the term is in $[\![\mathrm{t}]\!]_g \setminus [\![\mathrm{t}']\!]_g$ or $[\![\mathrm{t}']\!]_g \setminus [\![\mathrm{t}]\!]_g$. The procedure then can then be split into two cases depends on whether the guessed term is a guarded term or $\top$:

Theorem 10. *Deciding equality of TopKAT terms are PSPACE-complete.*

Proof. We first restrict the alphabet to only the primitives appeared in t and t'; then non-deterministically guess a term in $GT_{\top}$, and try to decide whether the term is in $[\![\mathsf{t}]\!]_g \setminus [\![\mathsf{t}']\!]_g$ or $[\![\mathsf{t}']\!]_g \setminus [\![\mathsf{t}]\!]_g$.

- If the term we guessed is $\top$, we can use the algorithm in Lemma 4 to decide wether $\top$ is in $[\![\mathsf{t}]\!]_g$ and $[\![\mathsf{t}']\!]_g$.
- If the term we guessed is a guarded term $\alpha_0 \mathsf{p}_1 \alpha_1 \dots \mathsf{p}_m \alpha_m$, we can use the PSPACE algorithm [Cohen et al. 1999, Section 5] to decide whether the term is in $[\![k(\mathsf{t})]\!]_g$ and $[\![k(\mathsf{t}')]\!]_g$. Since our alphabet only contains term appeared in t and t', the size of $k(\mathsf{t})$ and $k(\mathsf{t}')$ is polynomial to the size of t and t', hence the decision procedure is PSPACE to the size of t and t'. And by Lemma 3, we can also decide whether the guarded string is in $[\![\mathsf{t}]\!]_g$ and $[\![\mathsf{t}']\!]_g$.

If the term we guessed is in $[\![\mathsf{t}]\!]_g$ but not in $[\![\mathsf{t}']\!]_g$ (or in $[\![\mathsf{t}']\!]_g$ but not in $[\![\mathsf{t}]\!]_g$), we can conclude **TopKATs** $\models \mathsf{t} \neq \mathsf{t}'$, otherwise we conclude **TopKATs** $\models \mathsf{t} = \mathsf{t}'$. This algorithm can be made deterministic by Savitch's Theorem.

□

4.5 Equational Expressiveness of General Relational TopKATs

Given the completeness of general relational TopKAT, it is natural to wonder whether we can encode incorrectness logic in **TopGRELs**, so that the incompleteness of **TopRELs** is no longer a problem to reason about incorrectness logic in the theory of TopKAT.

However, we noticed the formulation of codomain will no longer work in **TopGRELs**. Recall the formulation of domain in **TopRELs**:

$$\top p = \top q \iff \mathrm{cod}(p) = \mathrm{cod}(q).$$

We take the **TopKATs** in Example 1, and let $p = \{(0,1)\}, q = \{(1,1)\}$, then we have $\mathrm{cod}(p) = \mathrm{cod}(q)$ but

$$\top p = \{(0,1),(0,0),(1,1)\} \fatsemi \{(0,1)\} = \{(0,1)\}$$
$$\top q = \{(0,1),(0,0),(1,1)\} \fatsemi \{(1,1)\} = \{(0,1),(1,1)\},$$

hence $\top p \neq \top q$, thus $\mathrm{cod}(p) = \mathrm{cod}(q) \implies \top p = \top q$ no longer holds.

Using the same method, we can also show that the formulation of incorrectness triple no longer holds in general relational TopKATs. Let $b = \{(0,0),(1,1)\}, p = \{(0,1)\}, c = \{(1,1)\}$, then

- the incorrectness triple $[b]\ p\ [c]$ holds.
- $\top bp = \top p = \{(0,1)\}$ but $c = \{(1,1)\}$, hence $\top bp \not\geq c$.

Therefore $\top bp \geq c \iff [b]\ p\ [c]$ no longer holds in general relational TopKATs.

It would be interesting to see if there exists another way to express incorrectness logic in general relational TopKATs. Unfortunately, we will see that not only it is not possible to express incorrectness in general relational TopKATs, but general relational TopKATs have the same equational expressiveness as **RELs**: any predicate expressible in general relational TopKATs can already be expressed in **RELs**.

Definition 18 (Equational Expressiveness of TopKATs)**.** *Given an alphabet K, B, a TopKAT $\mathcal{K}$, and an n-ary predicate $P : \mathcal{K}^n \to \mathbb{B}$, we say two terms $\mathsf{t}_1, \mathsf{t}_2 \in \mathrm{TopKATTerm}_{K,B}$* express *the predicate P over primitives $\mathsf{p}_1, \dots, \mathsf{p}_n \in K \cup B$ in $\mathcal{K}$, if for all valuations $u : K \cup B \to \mathcal{K}$:*

$$\models_u \mathsf{t}_1 = \mathsf{t}_2 \iff P([\![\mathsf{p}_1]\!]_u, [\![\mathsf{p}_2]\!]_u, \dots [\![\mathsf{p}_n]\!]_u)$$

A predicate is expressible in **TopGRELs** *if there exists a pair of TopKAT terms that express the predicate in all general relational TopKATs.*

Then we show that **TopGRELs** has the same equational expressiveness as **RELs**. Intuitively, the proof of equiexpressiveness exploits the fact that we can "simulate" the $\top$ term using the star of the sum of the entire alphabet. Thus, given two TopKAT terms that can express a predicate, we can construct two KAT terms, where $\top$ is simulated as above, to express the same predicate.

Lemma 5. *Given an alphabet K, B, and a pair of term $\mathsf{t}_1, \mathsf{t}_2 \in \mathrm{TopKATTerm}_{K,B}$, there exists a pair of term $\hat{\mathsf{t}_1}, \hat{\mathsf{t}_2} \in \mathrm{KATTerm}_{K,B}$, s.t. for all relational KATs $\mathcal{R}$ over X and relational valuation $u : K \cup B \to \mathcal{R}$, there exists a general relational TopKAT $\hat{\mathcal{R}}$ over X and valuation $\hat{u} : K \cup B \to \hat{\mathcal{R}}$ that is point-wise equal to u, i.e.*

$$\forall \mathsf{p} \in K \cup B,\ \hat{u}(\mathsf{p}) = u(\mathsf{p})$$

s.t. the following hold

$$[\![t]\!]_{\hat{u}} = [\![\hat{\mathsf{t}}]\!]_u$$

Theorem 11 (Equational Expressiveness of General Relational TopKATs)**.** *Given an alphabet K, B, a n-ary predicate P, the predicate P over primitives $\mathsf{p}_1, \mathsf{p}_2, \ldots, \mathsf{p}_n$ is expressible in* **TopGRELs** *iff they are expressible in* **RELs**.

Proof. Since general relational TopKATs are a subclass of **RELs**, if two KAT terms $\mathsf{t}_1 = \mathsf{t}_2$ express a predicate in **RELs**, the same pair of terms will express the same predicate in general relational TopKATs.

Then we show the other direction: if the predicate is expressible in general relational TopKATs, then it is expressible in **RELs**. By Lemma 5, if $\mathsf{t}_1 = \mathsf{t}_2$ express the predicate in **TopGRELs**, we construct $\hat{\mathsf{t}_1}$ and $\hat{\mathsf{t}_2}$; and for all relational valuations $u : K \cup B \to \mathcal{R}$, we find the general relational valuation $\hat{u} : K \cup B \to \hat{\mathcal{R}}$. Since $\mathsf{t}_1 = \mathsf{t}_2$ expresses P over $\mathsf{p}_1, \mathsf{p}_2, \ldots, \mathsf{p}_n$ in **TopGRELs**, we have

$$\models_{\hat{u}} \mathsf{t}_1 = \mathsf{t}_2 \iff P([\![\mathsf{p}_1]\!]_{\hat{u}}, [\![\mathsf{p}_2]\!]_{\hat{u}}, \ldots [\![\mathsf{p}_n]\!]_{\hat{u}}).$$

By Lemma 5, we know that

$$\forall \mathsf{p} \in K \cup B,\ [\![\mathsf{p}]\!]_{\hat{u}} = [\![\mathsf{p}]\!]_u,$$

thus

$$P([\![\mathsf{p}_1]\!]_{\hat{u}}, [\![\mathsf{p}_2]\!]_{\hat{u}}, \ldots [\![\mathsf{p}_n]\!]_{\hat{u}}) \iff P([\![\mathsf{p}_1]\!]_u, [\![\mathsf{p}_2]\!]_u, \ldots [\![\mathsf{p}_n]\!]_u).$$

Also by Lemma 5, we have $[\![\mathsf{t}_1]\!]_{\hat{u}} = [\![\hat{\mathsf{t}_1}]\!]_u$ and $[\![\mathsf{t}_2]\!]_{\hat{u}} = [\![\hat{\mathsf{t}_2}]\!]_u$, hence

$$\models_u \hat{\mathsf{t}_1} = \hat{\mathsf{t}_2} \iff \models_{\hat{u}} \mathsf{t}_1 = \mathsf{t}_2.$$

Finally, we conclude: for all relational valuations u,

$$\models_u \hat{\mathsf{t}_1} = \hat{\mathsf{t}_2} \iff \models_{\hat{u}} \mathsf{t}_1 = \mathsf{t}_2 \iff P([\![\mathsf{p}_1]\!]_{\hat{u}}, [\![\mathsf{p}_2]\!]_{\hat{u}}, \ldots [\![\mathsf{p}_n]\!]_{\hat{u}}) \iff P([\![\mathsf{p}_1]\!]_u, [\![\mathsf{p}_2]\!]_u, \ldots [\![\mathsf{p}_n]\!]_u)$$

Therefore $\hat{\mathsf{t}_1} = \hat{\mathsf{t}_2}$ expresses the predicate P over $\mathsf{p}_1, \mathsf{p}_2, \ldots, \mathsf{p}_n$ in **RELs**. □

Corollary 3. *General relational TopKATs cannot express incorrectness logic.*

5 MODELING ERRORS IN INCORRECTNESS LOGIC

One of the advantage of algebraic method is the ease of extension. In this section, we show how extending TopKAT with failure can naturally give rise to incorrectness triples that express abnormal termination. The main difference arises from the short-circuiting behavior of sequencing with errors. This can be seen in the following rule for sequential composition, which states that if an error already occurred in p, q will not be executed.

$$\frac{[b]\ p\ [er : c]}{[b]\ pq\ [er : c]}\ \textsc{Composition-Fail}.$$

To capture this type of control flow, we adapt the ideas from Mamouras [2017, Definition 3], who investigated similar issues in the setting of KAT.

Definition 19 (FailTopKAT)**.** *A FailTopKAT is a tuple* $(\mathcal{F}, \mathcal{K}, \mathcal{B}, \texttt{Fail})$*, where* $(\mathcal{K}, \mathcal{B})$ *is a TopKAT,* $\texttt{Fail} \in \mathcal{F}$*, and* $\mathcal{K} \subsetneq \mathcal{F}$*. The set* $\mathcal{F}$ *has the structure of a KAT that extends that of* $\mathcal{K}$*, except that the right annihilation rule* $p \cdot 0 = 0$ *need not hold. Instead,*

$$\texttt{Fail} \cdot p = \texttt{Fail},$$

where p *is any element of* $\mathcal{F}$*. (Crucially, we do* not *assume* $\top \geq \texttt{Fail}$*.) We call* $(\mathcal{K}, \mathcal{B})$ *the* fail-free subalgebras*, which model programs that do not fail. We will omit some of* $\mathcal{K}, \mathcal{B}, \texttt{Fail}$*, if they are not used or can be inferred from the context.*

The class of all FailTopKATs is denoted **FailTopKATs**.

Note that the original definition of Mamouras [2017] allows for try-catch statements and different types of errors. We omit these features for simplicity, since they are not needed in incorrectness logic [O'Hearn 2020].

There exists a canonical procedure for extending a TopKAT $\mathcal{K}$ with failures. The idea, which we adapt from Construction F [Mamouras 2017, Definition 4] is to consider elements of the form $(p, p') \in \mathcal{K} \times \mathcal{K}$, where p represents executions that terminate normally, and p' represents executions that fail.

Definition 20 (Construction F for **FailTopKATs**)**.** *Given a TopKAT* $(\mathcal{K}, \mathcal{B})$*, we construct a FailTopKAT* $(\mathcal{F}, \mathcal{K}', \mathcal{B}')$*. The carrier sets are defined as*

$$\mathcal{F} \triangleq \mathcal{K} \times \mathcal{K} \qquad \mathcal{K}' \triangleq \mathcal{K} \times \{0_{\mathcal{K}}\} \qquad \mathcal{B}' \triangleq \mathcal{B} \times \{0_{\mathcal{K}}\},$$

where $0_{\mathcal{K}}$ *is the additive identity in* $\mathcal{K}$*. The operations of* $\mathcal{F}$ *are defined as follows*

$$\begin{aligned}
0_{\mathcal{F}} &\triangleq (0_{\mathcal{K}}, 0_{\mathcal{K}}) \\
1_{\mathcal{F}} &\triangleq (1_{\mathcal{K}}, 0_{\mathcal{K}}) \\
\top_{\mathcal{F}} &\triangleq (\top_{\mathcal{K}}, 0_{\mathcal{K}}) \\
(p, p')(q, q') &= (pq, p' + pq') \\
(p, p') + (q, q') &= (p + q, p' + q') \\
(p, p')^{\star} &= (p^{\star}, (p^{\star})p') \\
\texttt{Fail} &= (0_{\mathcal{K}}, 1_{\mathcal{K}}) \\
\overline{(b, 0_{\mathcal{K}})} &= (\overline{b}, 0_{\mathcal{K}}).
\end{aligned}$$

To develop some intuition for this construction, suppose that $\mathcal{K}$ is a relational TopKAT over X. There exists a canonical embedding of $\mathcal{F}$ in $\mathcal{P}(X \times X \times \{ok, er\})$ that identifies $(p, p') \in \mathcal{F}$ with the relation $r = p \times \{ok\} \cup p' \times \{er\} \subseteq X \times X \times \{ok, er\}$. Intuitively, $(x, y, \epsilon) \in r$ means that a program took the input state x to the output state y, and the bit $\epsilon \in \{ok, er\}$ signals whether an error has occurred. By looking at the definition of sequential composition under this reading, it says that we get an error either by getting an error when running the first command (p'), or if we successfully run the first command, but get an error when running the second (pq'); and we terminates normally only we sequentially execute p and then q.

The semantics of O'Hearn [2020, Fig. 4] follows the same pattern, except that he considered the cases *ok* and *er* in separate relations. We can merge them back into a tuple, for example the semantics of `skip`

$$(\{(x, x) \mid x \in X\}, \emptyset)$$

coincides with the multiplicative identity 1 of applying construction F to a relational TopKAT $\mathcal{K}$:

$$(1_{\mathcal{K}}, 0).$$

In the same way, the sequential composition in is multiplication, the choice operator is addition, the star operator is the Kleene star, and the `error()` command is `Fail`. Thus applying F construction on a relational TopKAT will capture the semantics of programs with abnormal termination:

Definition 21 (Relational FailTopKAT)**.** *A relational FailTopKAT is a FailTopKAT constructed by applying construction F to a relational TopKAT. The class of all relational FailTopKATs is denoted* **FTopRELs**.

To better understand how to encode a incorrectness triple using FailTopKAT, we propose a definition of incorrectness triple equivalent to the original one [O'Hearn 2020, Definition 1 and 4]:

Definition 22. *Given a relational FailTopKAT $(\mathcal{F}, \mathcal{B})$, $p \in \mathcal{F}$, and $b, c \in \mathcal{B}$, for a error code $\epsilon \in \{ok, er\}$ an incorrectness triple $[b]\ p\ [\epsilon : c]$ is valid if*

$$\mathrm{cod}(bp) \supseteq \mathrm{cod}(c \cdot \hat{\epsilon})$$

where

$$\hat{\epsilon} \triangleq \begin{cases} 1 & \text{if } \epsilon = ok \\ \texttt{Fail} & \text{if } \epsilon = er. \end{cases}$$

and the function $\mathrm{cod}(-)$ *is extended component-wise:*

$$\mathrm{cod}((r, q)) \triangleq (\mathrm{cod}(r), \mathrm{cod}(q)).$$

Following the development of Section 3, we can obtain a formulation of incorrectness triple with abnormal termination.

Theorem 12 (Relational Validity)**.** *For all relational FailTopKATs $(\mathcal{F}, \mathcal{B})$, and $p, q \in \mathcal{F}$, we have*

$$\top p = \top q \iff \mathrm{cod}(p) = \mathrm{cod}(q)$$
$$\top p \leq \top q \iff \mathrm{cod}(p) \subseteq \mathrm{cod}(q),$$

Corollary 4. *For all relational FailTopKATs $\mathcal{F}$, for all $p \in \mathcal{F}$ and for all tests $b, c \in \mathcal{B}$, and for all $\epsilon \in \{ok, er\}$, the following holds*

$$[b]\ p\ [\epsilon : c] \iff \top bp \geq (c \cdot \hat{\epsilon}).$$

where $\hat{\epsilon}$ is defined in Definition 22

Thus, we can generalize incorrectness logic with errors to an arbitrary FailTopKAT.

Definition 23 (Abstract Incorrectness Triple With Failure)**.** *Given a* **FailTopKATs** $(\mathcal{F}, \mathcal{K}, \mathcal{B})$, $b, c \in \mathcal{B}$, $p \in \mathcal{F}$ *and* $\epsilon \in \{ok, er\}$, *we define*

$$[b]\ p\ [\epsilon : c] \triangleq \top bp \geq c \cdot \hat{\epsilon},$$

where $\hat{\epsilon}$ is defined as in Definition 22.

Theorem 13 (Soundness of Incorrectness Logic Rules in **FailTopKATs**)**.** *The rules in Figure 7 are valid for any FailTopKAT $(\mathcal{F}, \mathcal{K}, \mathcal{B})$, $a, b, c \in \mathcal{B}$ and $p, q \in \mathcal{F}$.*

$$\textsc{Empty}\ \frac{}{[b]\ p\ [\epsilon : 0]} \qquad \textsc{Consequence}\ \frac{b \le b' \qquad [b]\ p\ [\epsilon : c] \qquad c' \le c}{[b']\ p\ [\epsilon : c']} \qquad \textsc{Disjunction}\ \frac{[b_1]\ p\ [\epsilon : c_1] \qquad [b_2]\ p\ [\epsilon : c_2]}{[b_1 + b_2]\ p\ [\epsilon : (c_1 + c_2)]}$$

$$\textsc{Identity}\ \frac{}{[b]\ 1\ [ok : b, er : 0]} \qquad \textsc{Composition-Fail}\ \frac{[a]\ p\ [er : b]}{[a]\ pq\ [er : b]} \qquad \textsc{Composition-Normal}\ \frac{[a]\ p\ [ok : b] \qquad [b]\ p\ [\epsilon : c]}{[a]\ pq\ [\epsilon : c]}$$

$$\textsc{Choice-Left}\ \frac{[b]\ p\ [\epsilon : c]}{[b]\ p + q\ [\epsilon : c]} \qquad \textsc{Choice-Right}\ \frac{[b]\ q\ [\epsilon : c]}{[b]\ p + q\ [\epsilon : c]} \qquad \textsc{Assume}\ \frac{}{[a]\ b\ [ok : ab, er : 0]} \qquad \textsc{Error}\ \frac{}{[b]\ \texttt{Fail}\ [er : b]}$$

$$\textsc{Iter-Zero}\ \frac{}{[b]\ p^\star\ [ok : b]} \qquad \textsc{Iter-NonZero}\ \frac{[b]\ p^\star p\ [\epsilon : c]}{[b]\ p^\star\ [\epsilon : c]} \qquad \textsc{Iter-Dependent}\ \frac{\forall n \in \mathbb{N},\ [b_n]\ p\ [ok : b_{n+1}]}{[b_0]\ p^\star\ [ok : \sup_{n \in \mathbb{N}} b_n]}$$

Fig. 7. Complete set of incorrectness logic proof rule with both normal and abnormal termination

6 EXAMPLES: REASONING USING TOPKAT AND FAILTOPKAT

In this section, we show some concrete examples of algebraic program reasoning. We take the assignment language of O'Hearn [2020, Fig. 2], and regard assignments as primitive actions and assume statements as primitive tests. The relational semantics of this language forms a relational FailTopKAT, and it also forms a relational TopKAT if we do not consider the `error()` command and the *er* post-condition.

Example 2 (Incorrect Absolute Value Procedure)**.** *Here is an "incorrect procedure" for finding the absolute value of x:*

$$\texttt{if}\ x < 0\ \texttt{then skip else}\ x := -x,$$

To have a correct procedure for computing the absolute value the condition of the if statement should be $x > 0$. We can use TopKat to show that every negative number is reachable by using the following incorrectness triple:

$$[x < 0]\ \texttt{if}\ x < 0\ \texttt{then skip else}\ x := -x\ [x < 0]$$

This triple can be proven using just the theory of TopKAT. First, we can unfold the if statement:

$$[x < 0]\ (x < 0)1 + \overline{(x < 0)}(x := -x)\ [x < 0]$$

Then convert the triple to TopKAT encoding:

$$\top(x < 0)((x < 0)1 + \overline{(x < 0)}(x := -x)) \ge (x < 0)$$

Finally, we prove the above inequality:

$$\begin{aligned}
&\top(x < 0)((x < 0)1 + \overline{(x < 0)}(x := -x)) \\
&\ge \top(x < 0)(x < 0)1 && (p + q) \ge p \\
&= \top(x < 0) && \textit{idempotency of test} \\
&\ge (x < 0) && \top \ge 1
\end{aligned}$$

Thus we have shown that

$$[x < 0]\ (x < 0)1 + \overline{(x < 0)}(x := -x)\ [x < 0]$$

is valid, and the non-desirable results in $x < 0$ can be reached. As this triple can be shown just using the equational theory of TopKAT, this triple can be automatically decided using the algorithm in Theorem 10.

O'Hearn motivated the under-approximate triple as a way to reason about incorrect program. However the under-approximation logic can have other use cases. For example, we can mix under-approximation and over-approximation triples to prove a certain post condition is the strongest (as in Hoare logic) without a relational semantics. We show this next.

Example 3 (Reasoning With Hoare And Incorrectness Logic)**.** *The assertion $x \geq 0$ is the strongest post condition of program* `while` $x < 0$ `do` $x := x + 1$ *with precondition* true.

We can show this by the following two triples:

$$[\text{true}]\ \texttt{while}\ x < 0\ \texttt{do}\ x := x + 1\ [x \geq 0];$$
$$\{\text{true}\}\ \texttt{while}\ x < 0\ \texttt{do}\ x := x + 1\ \{x \geq 0\}.$$

Because for all p in some TopKAT, $p^\star \geq 1$, the incorrectness triple can be shown as follow

$$\top 1((x < 0)(x := x + 1))^\star \overline{(x < 0)} \geq \top 1 1 \overline{(x < 0)} \geq (x \geq 0).$$

And because $\top \geq 1((x < 0)(x := x + 1))^\star$, then the Hoare triple can be shown

$$1((x < 0)(x := x + 1))^\star \overline{(x < 0)} \leq \top \overline{(x < 0)} = \top (x \geq 0)$$

Example 4 (Theorem Proving In Hoare And Incorrectness Logic)**.** *For all TopKATs $(\mathcal{K}, \mathcal{B})$, and $b, c \in \mathcal{B}$, $p \in \mathcal{K}$, if $c \geq b$, then following incorrectness and Hoare triples are valid*

$$[c]\ \texttt{while}\ b\ \texttt{do}\ p\ [\overline{b}]\ \textit{and}\ \{c\}\ \texttt{while}\ b\ \texttt{do}\ p\ \{\overline{b}\}$$

This example is a generalization of Example 3. If we have a while loop with condition b, and the precondition c is larger than $\overline{b}$, then $\overline{b}$ is the strongest post-condition (in the sense of Hoare logic).

We first show a proof in a relational setting, as a comparison to the algebraic proof.

- $\mathrm{cod}(c(bp)^\star\overline{b}) \supseteq \mathrm{cod}(\overline{b})$*: because on the right hand side, every output needs to go through the final check of $\overline{b}$, it will also be in* $\mathrm{cod}(\overline{b})$*;*
- $\mathrm{cod}(c(bp)^\star\overline{b}) \subseteq \mathrm{cod}(\overline{b})$*: if the input of the right hand side is in* $\mathrm{cod}(\overline{b})$*, it will not be filtered out by c, will not go into loop $(bp)^\star$, will not be filtered out by $\overline{b}$, and will be outputted unchanged. thus everything in* $\mathrm{cod}(\overline{b})$ *will be a output of $(bp)^\star\overline{b}$, hence in* $\mathrm{cod}(c(bp)^\star\overline{b})$*.*

Alternatively, we prove this example algebraically. Because $(bp)^\star \geq 1$ and $c \geq \overline{b}$:

$$\top c(bp)^\star\overline{b} \geq \top c 1 \overline{b} = \top \overline{b} \geq \overline{b},$$

and because $\top \geq \top c(bp)^\star$:

$$\overline{b} \leq \top \overline{b} \geq \top c(bp)^\star \overline{b}.$$

We can also show that the Hoare triple is also valid with Kozen's encoding:

$$c(bp)^\star\overline{b}(\overline{\overline{b}}) = c(bp)^\star \overline{b} b = c(bp)^\star 0 = 0.$$

The fact that we can also use Kozen's encoding to reach the same conclusion is not surprising, as we have shown that Kozen's encoding is equivalent to our encoding in all TopKAT.

Example 5 (Error In Loop)**.** *This example simulates a while loop where the body will encounter an error when $x \leq 0$, and it will do some useful computation p if it does not encounter the error:*

$$[\text{true}]\ \texttt{while}\ x \geq 0\ \texttt{do}\ \texttt{if}\ x \leq 0\ \texttt{then}\ \texttt{error}()\ \texttt{else}\ p\ [er : x = 0]$$

the incorrect loop condition will trigger the possible error in the loop body, which is undesirable.

To show this triple, we need to show the following FailTopKAT inequalities:

$$\top 1((x \geq 0)((x \leq 0)(\texttt{Fail}) + (\overline{x \leq 0})(p)))^\star \geq (x = 0)\,\texttt{Fail}$$

The proof is as follow:

$$\begin{aligned}
&\top 1((x \geq 0)((x \leq 0)(\texttt{Fail}) + (\overline{x \leq 0})(p)))^\star && \\
&\geq \top 1((x \geq 0)(x \leq 0)(\texttt{Fail}))^\star && q + r \geq q \\
&= \top 1(x \geq 0)(x \leq 0)(\texttt{Fail}) && q^\star \geq q \\
&= \top (x = 0)\,\texttt{Fail} && (x \geq 0 \wedge x \leq 0) = (x = 0) \\
&\geq (x = 0)\,\texttt{Fail} && \top \geq 1
\end{aligned}$$

Notice that in the second last step we used the fact

$$(x \geq 0 \wedge x \leq 0) = (x = 0).$$

We invoked the logical meaning of $x \geq 0$, $x \leq 0$, and $x = 0$. Thus we are purely using the theory of TopKAT.

We can use more than just logical implications. Since our encoding of incorrectness logic is conservative (relationally valid), all the proof rules of incorrectness logic will play nicely with algebraic reasoning. The next example demonstrates the mix of equational reasoning and assignment rule.

Example 6 (Assignment)**.** *We have the same program as Example 2 with the precondition changed to $x \geq 0$:*

$$[x > 0]\ \texttt{if}\ x < 0\ \texttt{then}\ \texttt{skip}\ \texttt{else}\ x := -x\ [x < 0].$$

Even though the assignment rule from Incorrectness Logic, as well as the one from Hoare Logic are not propositional, in the sense we use in this paper, we can use specific inequalities to represent specific instances of these rules. For example, we can use an instance represented by the following incorrectness triple

$$[x > 0]\ x := -x\ [x < 0],$$

This triple corresponds to the following TopKAT inequality:

$$\top (x > 0)(x := -x) \geq (x < 0).$$

With the above inequality, we can derive the incorrectness triple in the example as follow:

$$\begin{aligned}
&\top (x > 0)((x < 0)1 + \overline{(x < 0)}(x := -x)) && \\
&\geq \top (x > 0)(\overline{(x < 0)}(x := -x)) && p + q \geq q \\
&\geq \top (x > 0)(x := -x) && (x > 0) \leq \overline{(x < 0)} \\
&\geq (x < 0) && \textit{above inequality}
\end{aligned}$$

Thus, as we can see, even though TopKAT does not contain assignments axiom, we can still reason about programs with assignments.

7 RELATED ENCODINGS IN KLEENE ALGEBRAS

The use of the complete relation as the top element in a relation-based algebraic structure traces back to the study of relation algebra [Andréka and Mikulás 2011; Maddux 1991]. However the extension of complete relation in to relational Kleene algebra was only recently studied in [Pous 2013, 2016], both articles provided the same counterexample to the completeness of relational Kleene algebra with top,

$$\mathsf{T}\mathsf{p}\mathsf{T}\mathsf{q}\mathsf{T} = \mathsf{T}\mathsf{q}\mathsf{T}\mathsf{p}\mathsf{T}.$$

However, our counterexamples $\mathsf{T}\mathsf{p} = \mathsf{T}\mathsf{p}\mathsf{T}\mathsf{p}$ and $\mathsf{p}\mathsf{T}\mathsf{p} \geq \mathsf{p}$ are simpler than Pous's. We also notice that $\mathsf{p}\mathsf{T}\mathsf{p} \geq \mathsf{p}$ is surprisingly similar to the counterexample for completeness of relational KAC$^-$ (Kleene algebra with converse) [Brunet and Pous 2014], which is

$$\mathsf{p}\mathsf{p}^\vee\mathsf{p} \geq \mathsf{p}.$$

Brunet and Pous [2014] solved the incompleteness problem by extending the equational system of KAC$^-$ with the counterexample $\mathsf{p}\mathsf{p}^\vee\mathsf{p} \geq \mathsf{p}$ obtaining the relationally complete system KAC.

To further expand on the potential connections between TopKAT and KAC, we have discover that the converse relation may also be able to express codomain. Our original formulation exploit the fact that for every relation p, $p^\vee p$ is larger than the identity relation on codomain of p ($\{(x,x) \mid x \in \mathrm{cod}(p)\}$), and smaller than the complete relation on codomain of p ($\{(x,y) \mid x,y \in \mathrm{cod}(p)\}$). Therefore, a relationally valid encoding for incorrectness triple $[b]\ p\ [c]$ can be

$$p^\vee bp \geq c.$$

several downsides of the KAT with converse encoding is the complicated equational theory of KAC and the lack of obvious way to formulate Hoare logic. However, this discovery still shines a light on the connection of KAC and codomain.

Contrary to the aforementioned attempt to capture the naive codomain of relations, there are other works that seek to have domain and codomain as a built-in operator in the algebraic theory [Desharnais et al. 2004, 2006; Fahrenberg et al. 2021].

In a concurrent recent work, Möller et al. [2021] showed an encoding of incorrectness logic in an extension of Kleene Algebra with a modal operator $\langle p|\, b$ aimed to model the strongest postcondition of the program p given the precondition b as a test. As in our work, they can also encode both Hoare and incorrectness triples:

$$[b]\ p\ [c] \triangleq \langle p|\, b \geq c,$$
$$\{b\}\ p\ \{c\} \triangleq \langle p|\, b \leq c.$$

Their algebra is called CTC (Countably Test Complete) Kleene Algebra, since they require all countable join of tests to exists, in order to obtain relative completeness of the incorrectness logic encoding.

8 RELATED WORKS

Kleene Algebra with Tests and extensions. The idea of Kleene Algebra with Tests was introduced by Kozen [1997] and its theory was studied in several subsequent works. Kozen and Smith [1997] focused on completeness and decidability of KAT equalities. In particular, they showed that KAT is complete over relational models and language models, also deciding equality of KAT terms is PSPACE complete by reduction to PDL. Cohen et al. [1999] gave a more elementary proof of the PSPACE complexity of deciding equality in KAT. In his seminal work, Kozen [2000] showed that KAT subsumes partial correctness of propositional Hoare logic. This result demonstrates the power of KAT in expressing program logics.

Inspired by earlier studies on relation algebra, Brunet and Pous [2014] developed an extension of Kleene Algebra with a converse relation. This system can also be used to express incorrectness logic, however it has a more complex equational theory than the one of TopKAT, which we present here. An extension of the equational theory with a top element was also considered in [Pous 2016] in the context of KA. This work showed the incompleteness of this extension over relational models. A top element was also used by Mamouras [2017] as a way to "forget the program state". The same work also extended KAT with failure to reason about abnormal termination. In a different direction, Anderson et al. [2014] extended KAT to NetKAT in order to provide a semantical foundation of network applications; and finally Smolka et al. [2017] further extended NetKAT to incorporate probabilistic reasoning. KAT has also been integrated into a Coq library by Pous [2013]. This library can be used to prove equivalences and correctness of while programs.

Incorrectness Logic and extensions. Incorrectness logic has been recently introduced by O'Hearn [2020] to reason about incorrect programs. O'Hearn was motivated by the practical need of providing proofs of failure and incorrectness. In his paper O'Hearn proposed a proof system for incorrectness logic and studied its underlying semantics model. A similar system to the one studied by O'Hearn was investigated by de Vries and Koutavas [2011] for reasoning about randomized algorithms. Murray [2020] implemented and formally verified a relational version of incorrectness logic in Isabelle. Murray logic is relational in the sense that it allows one to reason about two executions of two potentially different programs. It would be interesting to see if a similar logic could also be embedded naturally in TopKAT. Raad et al. [2020] combined incorrectness logic and separation logic to reason about incorrect programs in a local way without tracking the global state.

As we discussed in the previous section, in a recent concurrent work Möller et al. [2021] showed an encoding of incorrectness logic in an extension of Kleene Algebra with a modal operator for representing strongest postconditions. Our work differs from theirs in several aspects. First, we show the impossibility of encoding incorrectness logic in basic KAT. Second, we consider TopKat rather than adding modal operators. The two approaches share some similarities, especially in relational models, where they are essentially equivalent, as discussed in Möller et al. [2021]. Third, we do not require all countable join of test to exist, since we don't focus on relative completeness. On the other hand, we study the meta-theory of TopKAT in details.

9 CONCLUSION AND FUTURE WORK

We believe that our work has clarified the main questions about how to perform incorrectness reasoning in an equational algebraic system in the style of KAT. However, this work has also generated several other interesting questions. We discuss some of them here.

Completeness. We have shown that the equational theory of TopKAT is incomplete over relational TopKATs. This means that there might be valid incorrectness triples in relational TopKAT that cannot be validated by the equational theory of TopKAT. Hence, a natural open question is to investigate whether there are additional axioms that we could add to recover completeness over relational TopKATs. One way to approach this question is to consider an extension similar to the one studied by Brunet and Pous [2014] that we discussed in Section 7. It is natural to wonder if extending TopKAT with a similar rule can help to recover completeness over relational TopKATs.

Other directions. There are various abstractions of domain, namely [Desharnais et al. 2004, 2006; Fahrenberg et al. 2021], The conventional wisdom would suggest that these direct abstractions are more powerful than TopKAT (admits more models), but it would also be interesting to better understand the connections between the two. Fischer and Ladner [1979]; Kozen and Smith [1997]

showed strong connections between Kleene algebra with tests and propositional dynamic logic (PDL). It would be interesting to see how TopKAT would relate to propositional dynamic logic.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under award No. CNS 2040249 and Grant No. CNS 2040222. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

We thank Damien Pous, Alexandra Silva, Bernhard Möller, Peter O'Hearn, and all of our reviewers for their valuable inputs on this paper.

REFERENCES

Carolyn Jane Anderson, Nate Foster, Arjun Guha, Jean-Baptiste Jeannin, Dexter Kozen, Cole Schlesinger, and David Walker. 2014. NetKAT: semantic foundations for networks. *ACM SIGPLAN Notices* 49, 1 (Jan. 2014), 113–126. https://doi.org/10.1145/2578855.2535862

Hajnal Andréka and Szabolcs Mikulás. 2011. Axiomatizability of positive algebras of binary relations. *Algebra universalis* 66, 1-2 (Oct. 2011), 7–34. https://doi.org/10.1007/s00012-011-0142-3

Allegra Angus and Dexter Kozen. 2001. Kleene Algebra with Tests and Program Schematology. (July 2001). https://ecommons.cornell.edu/handle/1813/5831 Accepted: 2007-04-09T19:56:22Z Publisher: Cornell University.

Paul Brunet and Damien Pous. 2014. Kleene Algebra with Converse. In *Relational and Algebraic Methods in Computer Science*, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Alfred Kobsa, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Demetri Terzopoulos, Doug Tygar, Gerhard Weikum, Peter Höfner, Peter Jipsen, Wolfram Kahl, and Martin Eric Müller (Eds.). Vol. 8428. Springer International Publishing, Cham, 101–118. https://doi.org/10.1007/978-3-319-06251-8_7 Series Title: Lecture Notes in Computer Science.

Ernie Cohen, Dexter Kozen, and Frederick Smith. 1999. The Complexity of Kleene Algebra with Tests. (July 1999).

Edsko de Vries and Vasileios Koutavas. 2011. Reverse Hoare Logic. In *Software Engineering and Formal Methods*, Gilles Barthe, Alberto Pardo, and Gerardo Schneider (Eds.). Vol. 7041. Springer Berlin Heidelberg, Berlin, Heidelberg, 155–171. https://doi.org/10.1007/978-3-642-24690-6_12 Series Title: Lecture Notes in Computer Science.

Jules Desharnais, Bernhard Möller, and Georg Struth. 2004. Modal Kleene Algebra and Applications – A Survey. In *Journal on Relational Methods in Computer Science*. 93–131.

Jules Desharnais, Bernhard Möller, and Georg Struth. 2006. Kleene algebra with domain. *ACM Transactions on Computational Logic* 7, 4 (Oct. 2006), 798–833. https://doi.org/10.1145/1183278.1183285

Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiánski. 2021. Domain Semirings United. *arXiv:2011.04704 [cs]* (March 2021). http://arxiv.org/abs/2011.04704 arXiv: 2011.04704.

Michael J. Fischer and Richard E. Ladner. 1979. Propositional dynamic logic of regular programs. *J. Comput. System Sci.* 18, 2 (April 1979), 194–211. https://doi.org/10.1016/0022-0000(79)90046-1

Robert W. Floyd. 1967. Assigning Meanings to Programs. In *Mathematical Aspects of Computer Science*, J. T. Schwartz (Ed.). Symposia in Applied Mathematics, Vol. 19. American Mathematical Society, Providence, RI, 19–32.

Victor B. F. Gomes and Georg Struth. 2016. Modal Kleene Algebra Applied to Program Correctness. In *FM 2016: Formal Methods*, John Fitzgerald, Constance Heitmeyer, Stefania Gnesi, and Anna Philippou (Eds.). Vol. 9995. Springer International Publishing, Cham, 310–325. https://doi.org/10.1007/978-3-319-48989-6_19 Series Title: Lecture Notes in Computer Science.

C. A. R. Hoare. 1969. An Axiomatic Basis for Computer Programming. *Commun. ACM* 12, 10 (1969), 576–580. https://doi.org/10.1145/363235.363259

D. Kozen. 1994. A Completeness Theorem for Kleene Algebras and the Algebra of Regular Events. *Information and Computation* 110, 2 (May 1994), 366–390. https://doi.org/10.1006/inco.1994.1037

Dexter Kozen. 1997. Kleene algebra with tests. *ACM Transactions on Programming Languages and Systems* 19, 3 (May 1997), 427–443. https://doi.org/10.1145/256167.256195

Dexter Kozen. 2000. On Hoare logic and Kleene algebra with tests. *ACM Transactions on Computational Logic* 1, 1 (July 2000), 60–76. https://doi.org/10.1145/343369.343378

Dexter Kozen and Maria-Cristina Patron. 2000. Certification of Compiler Optimizations Using Kleene Algebra with Tests. In *Computational Logic — CL 2000 (Lecture Notes in Computer Science)*, John Lloyd, Veronica Dahl, Ulrich Furbach, Manfred Kerber, Kung-Kiu Lau, Catuscia Palamidessi, Luís Moniz Pereira, Yehoshua Sagiv, and Peter J. Stuckey (Eds.). Springer, Berlin, Heidelberg, 568–582. https://doi.org/10.1007/3-540-44957-4_38

Dexter Kozen and Frederick Smith. 1997. Kleene algebra with tests: Completeness and decidability. In *Computer Science Logic*, Gerhard Goos, Juris Hartmanis, Jan Leeuwen, Dirk Dalen, and Marc Bezem (Eds.). Vol. 1258. Springer Berlin Heidelberg, Berlin, Heidelberg, 244–259. https://doi.org/10.1007/3-540-63172-0_43 Series Title: Lecture Notes in Computer Science.

Roger D. Maddux. 1991. The origin of relation algebras in the development and axiomatization of the calculus of relations. *Studia Logica* 50, 3 (Sept. 1991), 421–455. https://doi.org/10.1007/BF00370681

Konstantinos Mamouras. 2017. Equational Theories of Abnormal Termination Based on Kleene Algebra. In *Foundations of Software Science and Computation Structures*, Javier Esparza and Andrzej S. Murawski (Eds.). Vol. 10203. Springer Berlin Heidelberg, Berlin, Heidelberg, 88–105. https://doi.org/10.1007/978-3-662-54458-7_6 Series Title: Lecture Notes in Computer Science.

Bernhard Möller, Peter O'Hearn, and Tony Hoare. 2021. On Algebra of Program Correctness and Incorrectness. In *Relational and Algebraic Methods in Computer Science*, Uli Fahrenberg, Mai Gehrke, Luigi Santocanale, and Michael Winter (Eds.). Vol. 13027. Springer International Publishing, Cham, 325–343. https://doi.org/10.1007/978-3-030-88701-8_20 Series Title: Lecture Notes in Computer Science.

Toby Murray. 2020. An Under-Approximate Relational Logic. *Archive of Formal Proofs* (Mar 2020).

Michael J. O'Donnell. 1985. *Equational Logic as a Programming Language*. The MIT Press.

Peter W. O'Hearn. 2020. Incorrectness logic. *Proceedings of the ACM on Programming Languages* 4, POPL (Jan. 2020), 1–32. https://doi.org/10.1145/3371078

Damien Pous. 2013. Kleene Algebra with Tests and Coq Tools for while Programs. In *Interactive Theorem Proving (Lecture Notes in Computer Science)*, Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie (Eds.). Springer, Berlin, Heidelberg, 180–196. https://doi.org/10.1007/978-3-642-39634-2_15

Damien Pous. 2016. *Automata for relation algebra and formal proofs*. Habilitation à diriger des recherches. ENS Lyon. https://hal.archives-ouvertes.fr/tel-01445821

V. R. Pratt. 1980. Dynamic algebras and the nature of induction. In *Proceedings of the twelfth annual ACM symposium on Theory of computing (STOC '80)*. Association for Computing Machinery, 22–28. https://doi.org/10.1145/800141.804649

Azalea Raad, Josh Berdine, Hoang-Hai Dang, Derek Dreyer, Peter O'Hearn, and Jules Villard. 2020. *Local Reasoning About the Presence of Bugs: Incorrectness Separation Logic*. Lecture Notes in Computer Science, Vol. 12225. Springer International Publishing, 225–252. https://doi.org/10.1007/978-3-030-53291-8_14

Steffen Smolka, Praveen Kumar, Nate Foster, Dexter Kozen, and Alexandra Silva. 2017. Cantor meets Scott: semantic foundations for probabilistic networks. *ACM SIGPLAN Notices* 52, 1 (Jan. 2017), 557–571. https://doi.org/10.1145/3093333.3009843

Walter Taylor. 1979. Equational logic. In *Universal Algebra* (second ed.), G. Grätzer (Ed.). Springer, New York, 378–400.

Cheng Zhang, Arthur Azevedo de Amorim, and Marco Gaboardi. 2021. On Incorrectness Logic and Kleene Algebra With Top and Tests. arXiv:2108.07707 [cs.PL]