Kleene Algebra with Tests: Completeness and Decidability

Dexter Kozen
kozen@cs.cornell.edu

Frederick Smith
fms@cs.cornell.edu

Computer Science Department
Cornell University
Ithaca, New York 14853-7501 USA

April 29, 1996

Abstract

Kleene algebras with tests provide a rigorous framework for equational specification and verification. They have been used successfully in basic safety analysis, source-to-source program transformation, and concurrency control. We prove the completeness of the equational theory of Kleene algebra with tests and *-continuous Kleene algebra with tests over language-theoretic and relational models. We also show decidability. Cohen's reduction of Kleene algebra with hypotheses of the form $r = 0$ to Kleene algebra without hypotheses is simplified and extended to handle Kleene algebras with tests.

1 Introduction

A *Kleene algebra with tests* is an algebraic structure consisting of a Kleene algebra with an embedded Boolean subalgebra. This formalism provides a rigorous framework for equational specification and verification of programs, and has been applied successfully to problems in basic safety analysis, source-to-source program transformation, and concurrency control [9, 10, 11, 21].

Kleene algebra with tests is based on Kleene algebra, a formalism that dates back to a 1956 paper of S. C. Kleene [16]. Kleene algebras have appeared in one form or another in relational algebra [26, 36], semantics and logics of programs [17, 30], automata and formal language theory [23, 24], and the design and analysis of algorithms [1, 15, 19]. Many authors have contributed over the years to the development of the theory [2, 3, 4, 5, 6, 7, 9, 12, 16, 17, 18, 20, 22, 24, 31, 32, 33, 34].

Kleene algebra forms an essential component of Propositional Dynamic Logic (PDL) [13], in which it is mixed with modal logic to give a theoretically appealing and practical system for reasoning about computation at the propositional level. Syntactically, PDL is a two-sorted logic consisting of *programs* and *propositions* defined by mutual induction. A basic operator in PDL is the *test operator* ?, by which a program φ? can be formed from any proposition φ. Intuitively, φ? acts as a guard that succeeds with no side effects in states satisfying φ and fails or aborts in states not satisfying φ. Tests are used to manipulate flow of control, and are needed to model conventional programming constructs such as conditionals and **while** loops.

From a practical standpoint, many simple program manipulations such as loop unwinding and basic safety analysis do not require the full power of PDL, but can be carried out in a purely equational subsystem using the axioms of Kleene algebra. However, tests are an essential ingredient for modeling real programs.

E. Cohen has studied Kleene algebra in the presence of extra conditions. Formally, these extra conditions take the form of premises in equational implications. He accommodates tests by defining a test to be an element b satisfying the condition $b \leq 1$. He has given several practical examples of the use of Kleene algebra with conditions in program verification, such as lazy caching and concurrency control [10, 11]. He has shown that Kleene algebra with extra conditions of the form $r = 0$ reduces to Kleene algebra without extra conditions [9], and is therefore decidable; but that *-continuous Kleene algebra in the presence of extra commutativity conditions of the form $pq = qp$, even for atomic p and q, is undecidable [8] (see [21] for a proof).

The equational axiomatization of Kleene algebra with tests (KAT) was introduced in [21]. In that paper we illustrated the utility of KAT by giving a purely equational proof of the following classical result: every **while** program can be simulated by a **while** program with at most one **while** loop [14, 25].

The proof in [21] needed extra commutativity conditions of the form

$bp = pb$, where b is a test. But as shown in that paper, this equation is equivalent to $bp\bar{b} + \bar{b}pb = 0$. Thus if Cohen's reduction of Kleene algebra with extra conditions $r = 0$ to Kleene algebra without extra conditions could be carried over to Kleene algebra with tests, then one would have effectively shown that the system used in [21] reduces to Kleene algebra with tests without extra conditions.

In this paper, we prove the following results.

1. A Kleene algebra with tests is **-continuous* if its Kleene algebra is *-continuous, *i.e.* if it satisfies the extra infinitary condition

$$pq^*r \quad = \quad \sup_n pq^n r \ .$$

   The system KAT with this additional axiom is called KAT^*. We show that the equational theory of Kleene algebras with tests and the equational theory of *-continuous Kleene algebras with tests coincide. Thus no more equations are derivable in KAT^* than are already derivable in KAT.

2. We show that KAT is complete over relational models. In other words, an equation $p = q$ is true in all Kleene algebras with tests iff it is true in all relational Kleene algebras with tests. This implies decidability of the equational theory by an essentially trivial reduction to Propositional Dynamic Logic (PDL). Since PDL is decidable in deterministic exponential time [27, 29], so is the equational theory of KAT. In a subsequent paper, we will show that the problem is *PSPACE*-complete, thus is of the same complexity as Kleene algebra without tests.

3. We show that the equational theory of Kleene algebra with tests admits free language-theoretic models consisting of regular sets of "guarded strings". These models play the same role in Kleene algebra with tests that the regular sets play in Kleene algebra. This result is analogous to the completeness result of [20], which states that the regular sets over a finite alphabet Σ form the free Kleene algebra on generators Σ.

4. As mentioned above, Cohen [9] shows that Kleene algebra with extra conditions $r = 0$ reduces efficiently to Kleene algebra without conditions. We simplify Cohen's construction and generalize it to handle Kleene algebra with tests. Thus all the machinery used in [21] can be reduced to Kleene algebra with tests without extra conditions.

2 Kleene Algebra with Tests

A *Kleene algebra with tests* [21] is a Kleene algebra with an embedded Boolean subalgebra. Formally, it is a two-sorted structure

$$(\mathcal{K},\ \mathcal{B},\ +,\ \cdot,\ ^*,\ ^-,\ 0,\ 1)$$

where $^-$ is a unary operator defined only on $\mathcal{B}$, such that

- $\mathcal{B} \subseteq \mathcal{K}$,
- $(\mathcal{K},\ +,\ \cdot,\ ^*,\ 0,\ 1)$ is a Kleene algebra, and
- $(\mathcal{B},\ +,\ \cdot,\ ^-,\ 0,\ 1)$ is a Boolean algebra.

The elements of $\mathcal{B}$ are called *tests*. We reserve the letters p, q, r, s for arbitrary elements of $\mathcal{K}$ and a, b, c for tests. In PDL, a test would be written $b?$, but since we are using different symbols for tests we can omit the ?.

This deceptively simple definition actually carries a lot of information. The operators $+$, $\cdot$, 0, and 1 play two roles: when applied to arbitrary elements of $\mathcal{K}$, they refer to choice, composition, fail, and skip, respectively; and when applied to tests, they take on the addition meaning of join, meet, falsity, and truth, respectively. The coexistence of these two usages admits considerable economy of expression.

As is customary, we will omit $\cdot$, writing pq instead of $p \cdot q$. The precedence of the operators is $^- > ^* > \cdot > +$. Thus $p + qr^*$ should be parsed $p + q(r^*)$.

2.1 Kleene Algebra

There have been many competing axiomatizations of Kleene algebra. The formulation we adopt here (KA) is from [20]. Succinctly put, a *Kleene algebra* is an (additively) idempotent, (multiplicatively) commutative semiring under $+, \cdot, 0, 1$, satisfying the additional properties

$$\begin{aligned} 1 + pp^* &= p^* && (1)\\ 1 + p^*p &= p^* && (2)\\ q + pr \le r &\rightarrow p^*q \le r && (3)\\ q + rp \le r &\rightarrow qp^* \le r && (4) \end{aligned}$$

where $\leq$ refers to the natural partial order on $\mathcal{K}$:

$$p \leq q \quad \leftrightarrow \quad p + q = q \; .$$

The operation $+$ gives the supremum with respect to the natural order $\leq$. Instead of (3) and (4), we might take the equivalent axioms

$$pr \leq r \quad \rightarrow \quad p^* r \leq r \tag{5}$$

$$rp \leq r \quad \rightarrow \quad rp^* \leq r \; . \tag{6}$$

These axioms say essentially that * behaves like the Kleene asterate operator of formal language theory or the reflexive transitive closure operator of relational algebra.

Typical models include the family of regular sets over a finite alphabet, the family of binary relations on a set, and the family of $n \times n$ matrices over another Kleene algebra.

A Kleene algebra is said to be *-*continuous* if it satisfies the infinitary condition

$$pq^*r \quad = \quad \sup_{n \geq 0} pq^n r \tag{7}$$

where

$$q^0 \quad \stackrel{\text{def}}{=} \quad 1$$

$$q^{n+1} \quad \stackrel{\text{def}}{=} \quad qq^n$$

and the supremum is with respect to the natural order $\leq$. We can think of (7) as a conjunction of the infinitely many equational axioms

$$pq^n r \quad \leq \quad pq^* r, \quad n \geq 0 \tag{8}$$

and the infinitary Horn formula

$$\bigwedge_{n \geq 0} pq^n r \leq s \quad \rightarrow \quad pq^* r \leq s \; . \tag{9}$$

In the presence of the other axioms, the *-continuity condition (7) implies (3)–(6), and is strictly stronger in the sense that there exist Kleene algebras that are not *-continuous [18].

The main result of [20] is that all true identities between regular expressions, interpreted as regular sets of strings, are derivable from the axioms of Kleene algebra [20], and only such identities are derivable. In other words, the algebra of regular sets of strings over the finite alphabet Σ is the free Kleene algebra on generators Σ. It is also the free *-continuous Kleene algebra on generators Σ; *i.e.*, the equational theory of the Kleene algebras and the *-continuous Kleene algebras coincide.

Two useful identities of Kleene algebra are

$$\begin{aligned} p^*(qp^*)^* &= (p+q)^* && (10)\\ p(qp)^* &= (pq)^*p\ . && (11)\end{aligned}$$

All the operators are monotone with respect to $\leq$. In other words, if $p \leq q$, then $pr \leq qr$, $p+r \leq q+r$, and $p^* \leq q^*$ for any r.

See [20] for a more thorough introduction.

2.2 The Boolean Subalgebra

The Boolean subalgebra $\mathcal{B}$ admits a Boolean negation operator $^-$ defined only on $\mathcal{B}$. Join and meet are given by the Kleene algebra operators $+$ and $\cdot$, respectively. That $\mathcal{B}$ is a Boolean algebra is asserted by the axioms

$$\begin{aligned} b &\leq 1\\ b+\overline{b} &= 1\\ b\overline{b} &= 0\\ bb &= b\\ bc &= cb \end{aligned}$$

in addition to the Kleene algebra axioms given above.

The sequential composition operator $\cdot$ acts as conjunction when applied to tests, and the choice operator $+$ acts as disjunction. Intuitively, a test bc succeeds iff both b and c succeed, and $b+c$ succeeds iff either b or c succeeds.

One of the Boolean algebra axioms states that $b \leq 1$ for all $b \in \mathcal{B}$. It is tempting to define tests in an arbitrary Kleene algebra $\mathcal{K}$ to be the set $\{p \in \mathcal{K} \mid p \leq 1\}$. This is the approach taken by Cohen [9]. Although it makes sense in algebras of binary relations, in general the set $\{p \in \mathcal{K} \mid p \leq 1\}$

may not extend to a Boolean algebra. For example, $p \leq 1$ for all p in the (min,+) Kleene algebra of the theory of algorithms, but the multiplicative idempotence law $pp = p$ fails.

Even over algebras of binary relations, we would like to admit models with programs whose input/output relations are subsets of the identity (*i.e.*, have no side effects) but whose complements are nevertheless uncomputable. We intend tests b to be viewed as simple predicates that are easily recognizable as such, and that are immediately decidable in a given state (and whose complements are therefore also immediately decidable).

We remark that under our definition, every Kleene algebra extends trivially to a Kleene algebra with tests by taking $\mathcal{B}$ to be the two-element Boolean algebra consisting of 0 and 1.

2.3 The Language of Kleene Algebra with Tests

Let Σ and B be disjoint finite sets of symbols. Elements of Σ are called *primitive actions* and elements of B are called *primitive tests*. *Terms* and *Boolean terms* are defined inductively:

- any primitive action p is a term
- any primitive test b is a Boolean term
- 0 and 1 are Boolean terms
- if p and q are terms, then so are $p+q$, pq, and p^* (suitably parenthesized if necessary)
- if b and c are Boolean terms, then so are $b + c$, bc, and $\overline{b}$ (suitably parenthesized if necessary)
- any Boolean term is a term.

The set of all terms over Σ and B is denoted $T_{\Sigma,\mathsf{B}}$. The set of all Boolean terms over B is denote T_{B}.

An *interpretation* over a Kleene algebra with tests $\mathcal{K}$ is any homomorphism (function commuting with the distinguished operations and constants) defined on $T_{\Sigma,\mathsf{B}}$ and taking values in $\mathcal{K}$ such that the Boolean terms are mapped to elements of the distinguished Boolean subalgebra. If $\mathcal{K}$ is a Kleene algebra with tests, we write $\mathcal{K}, I \models \varphi$ if the formula φ holds in $\mathcal{K}$ under the interpretation I according to the usual semantics of first-order logic. In this paper the only formulas we consider are equations or equational implications.

We write $\mathsf{KAT} \models \varphi$ if the formula φ is a logical consequence of KAT, *i.e.* if φ holds under all interpretations over Kleene algebras with tests. We write $\mathsf{KAT}^* \models \varphi$ if φ holds under all interpretations over *-continuous Kleene algebras with tests.

The following result was proved in [21].

Lemma 1 *In any Kleene algebra with tests, the following are equivalent:*

(i) $pb = bp$

(ii) $p\overline{b} = \overline{b}p$

(iii) $bp\overline{b} + \overline{b}pb = 0$.

Of course, any pair of tests always commute, *i.e.*, $bc = cb$; this is an axiom of Boolean algebra.

3 A Language-Theoretic Model

Let Σ and B be disjoint finite sets of symbols. Our language-theoretic model of Kleene algebras with tests is based on the idea of *guarded strings* over Σ and B. We obtain a guarded string from a string $x \in \Sigma^*$ by inserting *atoms* interstitially among the symbols of x. An *atom* is a Boolean expression representing an atom (minimal nonzero element) of the free Boolean algebra on generators B.

Formally, an *atom* of $\mathsf{B} = \{b_1, \ldots, b_k\}$ is a string of literals $c_1c_2 \cdots c_k$, where each $c_i \in \{b_i, \overline{b}_i\}$. This assumes an arbitrary but fixed order $b_1 < b_2 < \cdots < b_k$ on B; for technical reasons, we require the literals in an atom to occur in this order. There are exactly 2^k atoms. We denote atoms of B by $\alpha, \beta, \alpha_0, \ldots$ The set of all atoms of B is denoted $1_{\mathcal{G}}$ (this notation is chosen because $1_{\mathcal{G}}$ will turn out to be the multiplicative identity of our language-theoretic model $\mathcal{G}$).

If $b \in \mathsf{B}$ and α is an atom of B, we write $\alpha \leq b$ if b occurs positively in α and $\alpha \leq \overline{b}$ if b occurs negatively in α. This notation is consistent with the natural order in the free Boolean algebra generated by B.

Intuitively, the symbols of Σ can be thought of as instructions and atoms as conjunctions of conditions that must be satisfied at that point in the computation. If $\alpha \le c_i$, then α asserts that c_i holds (and $\overline{c}_i$ fails) at that point in the computation.

Definition 2 A *guarded string* over Σ and B is any element of $(1_{\mathcal{G}}\Sigma)^*1_{\mathcal{G}}$, *i.e.*, any string

$$\alpha_0 p_1 \alpha_1 p_2 \cdots p_n \alpha_n \ , \quad n \ge 0 \ ,$$

where each α_i is an atom of B and each $p_i \in \Sigma$. Note that a guarded string begins and ends with an atom. In the case $n = 0$, a guarded string is just a single atom.

The set of all guarded strings over Σ and B is denoted $\mathrm{GS}_{\Sigma,\mathsf{B}}$, or just GS when Σ and B are understood. $\Box$

We denote strings in $(\Sigma \cup \mathsf{B})^*$, including guarded strings, by the letters $x, y, z, x_1, \ldots$.

The analog of concatenation for guarded strings is *coalesced product* ($\diamond$).

Definition 3 The *coalesced product* operation $\diamond$ is a *partial* binary operation on GS defined as follows:

$$x\alpha \diamond \beta y \ \stackrel{\mathrm{def}}{=} \ \begin{cases} x\alpha y \ , & \text{if } \alpha = \beta \\ \text{undefined} \ , & \text{otherwise.} \end{cases}$$

In other words, if the terminal atom of the first string is the same as the initial atom of the second string, then the two strings can be *coalesced*. This is like concatenation, except that we combine the two intermediate atoms into one.

If $A, B \subseteq \mathrm{GS}$, then

$$A \diamond B \ \stackrel{\mathrm{def}}{=} \ \{x \diamond y \mid x \in A, \ y \in B\} \ .$$

Thus $A \diamond B$ consists of all existing coalesced products of guarded strings in A with guarded strings in B. $\Box$

Example 4 Let $\mathcal{B} = \{b, c\}$ and $\Sigma = \{p, q, r, s\}$. Let

$$\begin{aligned} A &= \{bcp\overline{b}\overline{c}, \overline{b}\overline{c}, bcqb\overline{c}\} \\ B &= \{\overline{b}\overline{c}rbc, \overline{b}\overline{c}, b\overline{c}sbc\} \ . \end{aligned}$$

Then

$$A \diamond B \;=\; \{bcp\overline{b}\overline{c}rbc,\, bcp\overline{b}\overline{c},\, \overline{b}\overline{c}rbc,\, \overline{b}\overline{c},\, bcqb\overline{c}sbc\}\ .$$

$\square$

Note that the identity for coalesced product is $1_{\mathcal{G}}$.

Whereas the operation $\diamond$ is partial when applied to guarded strings, it is total when applied to *sets* of guarded strings. If there are no existing coalesced products of strings from A and B, then $A \diamond B = \varnothing$. It is not difficult to show that $\diamond$ is associative and that coalesced product distributes over union.

We now define a language-theoretic model $\mathcal{G} = \mathcal{G}_{\Sigma,\mathsf{B}}$ based on guarded strings. The elements of $\mathcal{G}$ will be the regular sets of guarded strings over Σ and B (although we have not yet defined *regular* in this context). We will also give a standard interpretation of terms in $T_{\Sigma,\mathsf{B}}$ over $\mathcal{G}$ analogous to the standard interpretation of regular expressions as regular sets.

For $A \subseteq \mathrm{GS}$, define inductively

$$\begin{aligned} A^0 &\stackrel{\text{def}}{=} 1_{\mathcal{G}} \\ A^{n+1} &\stackrel{\text{def}}{=} A \diamond A^n\ . \end{aligned}$$

The asterate operation for sets of guarded strings is defined by

$$A^* \stackrel{\text{def}}{=} \bigcup_{n \geq 0} A^n\ .$$

Let $^-$ denote set complementation in $1_{\mathcal{G}}$. That is, if $A \subseteq 1_{\mathcal{G}}$, then $\overline{A} = 1_{\mathcal{G}} - A$. Consider the structure

$$\mathcal{P}_{\Sigma,\mathsf{B}} \;=\; (2^{\mathrm{GS}},\ 2^{1_{\mathcal{G}}},\ \cup,\ \diamond,\ ^*,\ ^-,\ \varnothing,\ 1_{\mathcal{G}})\ .$$

We write $\mathcal{P}$ for $\mathcal{P}_{\Sigma,\mathsf{B}}$ when Σ and B are understood. It is quite straightforward to verify that $\mathcal{P}$ is a *-continuous Kleene algebra with tests, *i.e.* is a model of KAT^*. The Boolean algebra axioms hold for $2^{1_{\mathcal{G}}}$ because it is a set-theoretic Boolean algebra.

The *-continuity condition follows immediately from the definition of * and the distributivity of coalesced product over infinite union. Since

$$B^* \;=\; \bigcup_{n \geq 0} B^n\ ,$$

we have that

$$\begin{aligned} A \diamond B^* \diamond C &= A \diamond (\bigcup_{n\geq 0} B^n) \diamond C \\ &= \bigcup_{n\geq 0} A \diamond B^n \diamond C \ . \end{aligned}$$

Both of these expressions denote the set

$$\{x \diamond y \diamond z \mid x \in A,\ z \in C,\ \exists n\ y \in B^n\} \ .$$

The structure $\mathcal{G} = \mathcal{G}_{\Sigma,\mathsf{B}}$ is defined to be the subalgebra of $\mathcal{P}$ generated by the elements

$$\begin{array}{ll} \{\alpha p \beta \mid \alpha, \beta \in 1_{\mathcal{G}}\} \ , & p \in \Sigma \\ \{\alpha \in 1_{\mathcal{G}} \mid \alpha \leq b\} \ , & b \in B \ . \end{array}$$

That is, it is the smallest subalgebra of $\mathcal{P}$ containing these sets and closed under all the operations. Elements of $\mathcal{G}$ are called *regular sets*.

3.1 Standard Interpretation

We now define a standard interpretation G over $\mathcal{G}$. The map G associates a set of guarded strings with each term in $T_{\Sigma,\mathsf{B}}$ and is defined inductively.

$$\begin{aligned} G(p) &= \{\alpha p \beta \mid \alpha, \beta \in 1_{\mathcal{G}}\} \ , \quad p \in \Sigma \\ G(b) &= \{\alpha \in 1_{\mathcal{G}} \mid \alpha \leq b\} \ , \quad b \in B \\ G(p+q) &= G(p) \cup G(q) \\ G(pq) &= G(p) \diamond G(q) \\ G(1) &= 1_{\mathcal{G}} \\ G(0) &= \varnothing \\ G(\overline{b}) &= 1_{\mathcal{G}} - G(b) \\ G(p^*) &= G(p)^* \ . \end{aligned}$$

In other words, the map $G : T_{\Sigma,\mathsf{B}} \rightarrow \mathcal{G}$ is the homomorphism uniquely determined by its values on primitive actions and primitive tests as specified in the first two lines above.

4 Relational Models

Relational Kleene algebras with tests are interesting because they closely model our intuition about programs. In a relational model, the elements of $\mathcal{K}$ are binary relations and $\cdot$ is interpreted as relational composition. Elements of the Boolean subalgebra are subsets of the identity relation.

Formally, a *relational Kleene algebra with tests* on a set X is any structure

$$(\mathcal{K},\ \mathcal{B},\ \cup,\ \circ,\ {}^*,\ {}^-,\ \varnothing,\ \iota)$$

such that

$$(\mathcal{K},\ \cup,\ \circ,\ {}^*,\ \varnothing,\ \iota)$$

is a relational Kleene algebra, *i.e.* $\mathcal{K}$ is a family of binary relations on X, $\circ$ is ordinary relational composition, * is reflexive transitive closure, and ι is the identity relation on X; and

$$(\mathcal{B},\ \cup,\ \circ,\ {}^-,\ \varnothing,\ \iota)$$

is a Boolean algebra of subsets of ι.

All relational Kleene algebras with tests are *-continuous. We write $\mathsf{REL} \models \varphi$ if the formula φ holds in all relational Kleene algebras in the usual sense of first-order logic.

5 Completeness of KAT^* under the Standard Interpretation

In this section we prove that an equation $p = q$ is a theorem of *-continuous Kleene algebra with tests iff it holds in the standard interpretation over $\mathcal{G}_{\Sigma,\mathsf{B}}$, where Σ and B contain all primitive action and test symbols, respectively, appearing in p and q. We will later strengthen this result in §7 by removing the assumption of *-continuity.

Theorem 5 *Let $p, q \in T_{\Sigma,\mathsf{B}}$. Then*

$$\mathsf{KAT}^* \models p = q \iff G(p) = G(q)\ .$$

Equivalently, $\mathcal{G}_{\Sigma,\mathsf{B}}$ is the free *-continuous Kleene algebra with tests on generators Σ and B.

The forward implication is easy, since $\mathcal{G}$ is a *-continuous Kleene algebra. The converse is a consequence of the following lemma.

Lemma 6 *For any *-continuous Kleene algebra with tests $\mathcal{K}$, interpretation $I : T_{\Sigma,\mathsf{B}} \to \mathcal{K}$, and $p, q, r \in T_{\Sigma,\mathsf{B}}$,*

$$I(pqr) \quad = \quad \sup_{x \in G(q)} I(pxr)$$

where the supremum is with respect to the natural order in $\mathcal{K}$. In particular,

$$I(q) \quad = \quad \sup_{x \in G(q)} I(x) \ .$$

This result is analogous to the same result for Kleene algebras [19, pp. 221, 246] and the proof is similar. Note that the *-continuity axiom is a special case.

We are most interested in the second statement, but there is a slight subtlety that requires the stronger first statement as the induction hypothesis. In addition to the existence of the supremum, the more general statement provides a kind of infinite distributivity law over existing suprema. The need for this arises mainly in the induction case for $\cdot$.

Proof of Lemma 6. We proceed by induction on the structure of q. The basis consists of cases for primitive tests, primitive actions, 0 and 1. We argue the case for primitive actions and primitive tests explicitly.

For a primitive action $q \in \Sigma$, recall that

$$G(q) \quad = \quad \{\alpha q \beta \mid \alpha, \beta \in 1_{\mathcal{G}}\} \ .$$

Then

$$\begin{aligned} I(pqr) &= I(p)I(1)I(q)I(1)I(r) \\ &= \sup\{I(p)I(\alpha)I(q)I(\beta)I(r) \mid \alpha, \beta \in 1_{\mathcal{G}}\} \\ &= \sup\{I(p\alpha q \beta r) \mid \alpha, \beta \in 1_{\mathcal{G}}\} \\ &= \sup\{I(pxr) \mid x \in G(q)\} \ . \end{aligned}$$

Finite distributivity was used in the second step.

For a primitive test $b \in B$, recall that

$$G(b) \quad = \quad \{\alpha \mid \alpha \leq b\} \ .$$

Then

$$\begin{aligned} I(pbr) &= I(p)I(b)I(r) \\ &= \sup\{I(p)I(\alpha)I(r) \mid \alpha \leq b\} \\ &= \sup\{I(p\alpha r) \mid \alpha \leq b\} \\ &= \sup\{I(pxr) \mid x \in G(b)\} \ . \end{aligned}$$

Again, finite distributivity was used in the second step.

The induction step consists of cases for $+$, $\cdot$, *, and $^-$. We argue the cases for $\cdot$ and $^-$ explicitly; the other cases are the same as in [19, pp. 221, 246].

For the case $\cdot$, recall that

$$\begin{aligned} G(qq') &= G(q) \diamond G(q') \\ &= \{y\alpha z \mid y\alpha \in G(q),\ \alpha z \in G(q')\} \ . \end{aligned}$$

Applying the induction hypothesis twice,

$$\begin{aligned} I(pqq'r) &= \sup\{I(pqvr) \mid v \in G(q')\} \\ &= \sup\{\sup\{I(puvr) \mid u \in G(q)\} \mid v \in G(q')\} \\ &= \sup\{I(puvr) \mid u \in G(q),\ v \in G(q')\} \ . \end{aligned}$$

The last step follows from a purely lattice-theoretic argument: if all the suprema in question on the left hand side exist, then the supremum on the right hand side exists and the two sides are equal.

Now

$$\begin{aligned} &\sup\{I(puvr) \mid u \in G(q),\ v \in G(q')\} \\ &\quad = \sup\{I(py\alpha\beta zr) \mid y\alpha \in G(q),\ \beta z \in G(q')\} \\ &\quad = \sup\{I(py\alpha\alpha zr) \mid y\alpha \in G(q),\ \alpha z \in G(q')\} \qquad (12) \\ &\quad = \sup\{I(py\alpha zr) \mid y\alpha \in G(q),\ \alpha z \in G(q')\} \\ &\quad = \sup\{I(pxr) \mid x \in G(qq')\} \ . \end{aligned}$$

The justification for step (12) is that if $\alpha \neq \beta$, then the product in $\mathcal{K}$ is 0 and does not contribute to the supremum.

For the case $^-$, recall that

$$\begin{aligned} G(\overline{b}) &= 1_{\mathcal{G}} - G(b) \\ &= \{\alpha \mid \alpha \not\leq b\} \\ &= \{\alpha \mid \alpha \leq \overline{b}\} \ . \end{aligned}$$

Then

$$\begin{aligned} I(p\overline{b}r) &= \sup\{I(p\alpha r) \mid \alpha \leq \overline{b}\} \\ &= \sup\{I(p\alpha r) \mid \alpha \in G(\overline{b})\} \ . \end{aligned}$$

□

Proof of Theorem 5. If $\mathsf{KAT}^* \models p = q$ then $G(p) = G(q)$, since $\mathcal{G}$ is a *-continuous Kleene algebra with tests. Conversely, if $G(p) = G(q)$, then by Lemma 6, for any *-continuous Kleene algebra with tests $\mathcal{K}$ and any interpretation I over $\mathcal{K}$, $I(p) = I(q)$. Therefore $\mathsf{KAT}^* \models p = q$. □

6 Completeness over Relational Models

In this section we establish completeness over relational models. It will suffice to construct a relational model isomorphic to $\mathcal{G}$. This construction is similar to a construction of Pratt [28] for regular sets.

For A any set of guarded strings, define

$$h(A) \stackrel{\text{def}}{=} \{(x, x \diamond y) \mid x \in \mathrm{GS},\ y \in A\} \ .$$

Lemma 7 *The language-theoretic model $\mathcal{P}$ and its submodel $\mathcal{G}$ are isomorphic to relational models.*

Proof. We show that the function $h : \mathcal{P} \to 2^{\mathrm{GS}\times\mathrm{GS}}$ defined above embeds $\mathcal{P}$ isomorphically onto a subalgebra of the Kleene algebra of all binary relations on GS.

First we show that h is a homomorphism.

$$\begin{aligned} h(A \cup B) &= h(A) \cup h(B) \\ h(A \diamond B) &= \{(z, z \diamond r) \mid z \in \mathrm{GS},\ r \in A \diamond B\} \end{aligned}$$

$$
\begin{aligned}
&= \{(z, z \diamond p \diamond q) \mid z \in \mathrm{GS},\ p \in A,\ q \in B\} \\
&= \{(z, z \diamond p) \mid z \in \mathrm{GS},\ p \in A\} \\
&\qquad \circ \{(z \diamond p, z \diamond p \diamond q) \mid z \in \mathrm{GS},\ p \in A,\ q \in B\} \\
&= \{(z, z \diamond p) \mid z \in \mathrm{GS},\ p \in A\} \\
&\qquad \circ \{(y, y \diamond q) \mid y \in \mathrm{GS},\ q \in B\} \\
&= h(A) \circ h(B) \\
h(A^*) &= h(\bigcup_{n \geq 0} A^n) \\
&= \bigcup_{n \geq 0} h(A)^n \\
&= h(A)^* \\
h(1_{\mathcal{G}}) &= \{(x, x \diamond \alpha) \mid x \in \mathrm{GS}, \alpha \in 1_{\mathcal{G}}\} \\
&= \{(x, x) \mid x \in \mathrm{GS}\} \\
&= \iota \\
h(0) &= \varnothing \\
h(\overline{B}) &= h(\{\alpha \mid \alpha \notin B\}) \\
&= \{(x, x \diamond \alpha) \mid \alpha \notin B\} \\
&= \{(y\alpha, y\alpha) \mid \alpha \notin B\} \\
&= \iota - \{(y\alpha, y\alpha) \mid \alpha \in B\} \\
&= \iota - h(B)\ .
\end{aligned}
$$

The function h is injective, since A is uniquely recoverable from $h(A)$:

$$
A \quad = \quad \{y \mid \exists\alpha\ (\alpha, y) \in h(A)\}\ .
$$

The submodel $\mathcal{G}$ is perforce isomorphic to a relational model on GS, namely the image of $\mathcal{G}$ under h. □

The following theorem establishes the completeness of KAT^* over relational models.

Theorem 8 *Let* REL *denote the class of all relational Kleene algebras with tests. Let* $p, q \in T_{\Sigma,\mathsf{B}}$. *The following are equivalent:*

(i) $\mathsf{KAT}^* \vDash p = q$

(ii) $G(p) = G(q)$

(iii) $\mathsf{REL} \models p = q$.

Proof. The equivalence of (i) and (ii) was proved in Theorem 5. Since all relational models are *-continuous Kleene algebras with tests, (i) implies (iii). Finally, (iii) implies (ii) by Lemma 7. $\square$

7 Completeness of KAT

In this section we show that the equational theories of the Kleene algebras with tests and the *-continuous Kleene algebras with tests coincide; *i.e.*, we can add the statement $\mathsf{KAT} \models p = q$ to the list of equivalent conditions in the statement of Theorem 8. Equivalently, $\mathcal{G}$ is the free Kleene algebra with tests on generators Σ and B. Thus, for the purpose of deriving identities, the infinitary *-continuity condition provides no extra power over the finitary axiomatization KAT. (However, it does entail more equational implications [21]).

One possible approach might be to extend the completeness proof of [20] for KA to handle tests. We take a different approach here. We show that every term p can be transformed to a KAT-equivalent term $\widehat{p}$ such that $G(\widehat{p})$, the set of guarded strings represented by $\widehat{p}$, is the same as $R(\widehat{p})$, the set of strings represented by $\widehat{p}$ under the ordinary interpretation of regular expressions. The Boolean algebra axioms are not needed in equivalence proofs involving such terms, so we can apply the completeness result of [20] directly.

Consider the set $\overline{\mathsf{B}} = \{\overline{b} \mid b \in \mathsf{B}\}$, the set of negated atomic tests. We can view $\overline{\mathsf{B}}$ as a separate set of primitive symbols disjoint from B and Σ. Using the DeMorgan laws and the law $\overline{\overline{b}} = b$ of Boolean algebra, every term p can be transformed to a KAT-equivalent term p' in which $^-$ is applied only to primitive test symbols, thus we can view p' as a regular expression over the alphabet $\Sigma \cup \mathsf{B} \cup \overline{\mathsf{B}}$. As such, it represents a set of strings

$$R(p') \quad \subseteq \quad (\Sigma \cup \mathsf{B} \cup \overline{\mathsf{B}})^*$$

under the standard interpretation R of regular expressions as regular sets.

In general, the sets $R(p')$ and $G(p')$ may differ. For a simple example, $R(q) = \{q\}$ for primitive action q, but $G(q) = \{\alpha q \beta \mid \alpha, \beta \in 1_{\mathcal{G}}\}$.

Our main task will be to show how to further transform p' to another KAT-equivalent string $\widehat{p}$ such that all elements of $R(\widehat{p})$ are guarded strings, and in fact $R(\widehat{p}) = G(\widehat{p})$. We can then use the completeness result of [20], since p and q will be KAT-equivalent iff $\widehat{p}$ and $\widehat{q}$ are equivalent as regular expressions over $\Sigma \cup \mathsf{B} \cup \overline{\mathsf{B}}$, *i.e.*, if they can be proved equivalent in pure Kleene algebra.

Example 9 Consider the two terms

$$\begin{aligned} p &= (q + b + \overline{b})^* br \\ \widehat{p} &= (bq + \overline{b}q)^* br(b + \overline{b}) \ , \end{aligned}$$

where $\Sigma = \{q, r\}$ and $\mathsf{B} = \{b\}$. There are certainly strings in $R(p)$, *e.g.* $qq\overline{b}bbqbr$, that are not guarded strings. However, p and $\widehat{p}$ represent the same set of guarded strings under the interpretation G, and all strings in $R(\widehat{p})$ are guarded strings; *i.e.*, $G(p) = G(\widehat{p}) = R(\widehat{p})$. $\square$

In our inductive proof, it will be helpful to maintain terms in the following special form. Call a term *externally guarded* if it is of the form α or $\alpha q \beta$, where α and β are atoms of B. Define the *coalesced product* of two such terms as follows:

$$r\alpha \diamond \beta s \stackrel{\text{def}}{=} \begin{cases} r\alpha s \ , & \text{if } \alpha = \beta, \\ 0 \ , & \text{if } \alpha \neq \beta. \end{cases}$$

(Here we must distinguish between a guarded string as a guarded string and a guarded string as a term, since coalesced product is undefined for incompatible pairs of guarded strings.)

For any two externally guarded terms q and r,

$$G(q \diamond r) = G(q) \diamond G(r) \ ,$$

and $q \diamond r$ is externally guarded.

If $\sum_i q_i$ and $\sum_j r_j$ are sums of zero or more externally guarded terms, define

$$(\sum_i q_i) \diamond (\sum_j r_j) \stackrel{\text{def}}{=} \sum_{i,j} q_i \diamond r_j \ .$$

For any two sums q and r of externally guarded terms,

$$G(q \diamond r) = G(q) \diamond G(r) \ ,$$

and $q \diamond r$ is a sum of externally guarded terms.

Lemma 10 *For every term p, there is a term $\widehat{p}$ such that*

(i) $\mathsf{KAT} \models p = \widehat{p}$

(ii) $R(\widehat{p}) = G(\widehat{p})$

(iii) *$\widehat{p}$ is a sum of zero or more externally guarded terms.*

Proof. As argued above, we can assume without loss of generality that all occurrences of $^{-}$ in p are applied to primitive tests only, thus we may view p as a term over the alphabet $\Sigma \cup \mathsf{B} \cup \overline{\mathsf{B}}$.

We define $\widehat{p}$ by induction on the structure of p. For the basis, take

$$\begin{aligned}
\widehat{p} &\stackrel{\text{def}}{=} \sum_{\alpha,\beta\in 1_G} \alpha p \beta \ , \quad p \in \Sigma \\
\widehat{b} &\stackrel{\text{def}}{=} \sum_{\alpha\le b} \alpha \ , \quad b \in \mathsf{B} \cup \overline{\mathsf{B}} \\
\widehat{1} &\stackrel{\text{def}}{=} \sum_{\alpha\in 1_G} \alpha \\
\widehat{0} &\stackrel{\text{def}}{=} 0 \ .
\end{aligned}$$

In each of these cases, it is straightforward to verify (i), (ii), and (iii).

For the induction step, suppose we have terms p and q satisfying (ii) and (iii). We take

$$\begin{aligned}
\widehat{p+q} &\stackrel{\text{def}}{=} p + q \\
\widehat{pq} &\stackrel{\text{def}}{=} p \diamond q \ .
\end{aligned}$$

These constructions are easily shown to satisfy (i), (ii), and (iii).

It remains to construct $\widehat{p^*}$. We proceed by induction on the number of externally guarded terms in the sum p.

For the basis, we define

$$\begin{aligned}
\widehat{0^*} &\stackrel{\text{def}}{=} \widehat{1} \\
\widehat{\alpha^*} &\stackrel{\text{def}}{=} \alpha \\
\widehat{(\alpha q \beta)^*} &\stackrel{\text{def}}{=} \widehat{1} + \alpha q \beta \ , \quad \alpha \ne \beta && (13) \\
\widehat{(\alpha q \alpha)^*} &\stackrel{\text{def}}{=} \alpha + \alpha q (\alpha q)^* \alpha \ . && (14)
\end{aligned}$$

For the induction step, let $p = q + r$, where r is an externally guarded term and q is a sum of externally guarded terms, one fewer in number than in p. By the induction hypothesis, we can construct $q' = \widehat{q^*}$ with the desired properties. Suppose the initial atom of the externally guarded term r is α. Then $\mathsf{KAT} \models r = \alpha r$. Moreover, the expression $(rq'\alpha)^*$ is KAT-equivalent to $(r \diamond q' \diamond \alpha)^*$, which by distributivity can be put into a form in which (13) or (14) applies, yielding a term q'' satisfying (ii) and (iii).

Reasoning in KAT,

$$
\begin{aligned}
p^* &= (q + r)^* \\
&= q^*(rq^*)^* && \text{by (10)} \\
&= q'(rq')^* \\
&= q' + q'rq'(rq')^* && \text{by (1) and distributivity} \\
&= q' + q'rq'(\alpha rq')^* \\
&= q' + q'(rq'\alpha)^* rq' && \text{by (11)} \\
&= q' + q'q''rq' \\
&= q' + q' \diamond q'' \diamond r \diamond q' \ ,
\end{aligned}
$$

which is of the desired form. □

Theorem 11

$$\mathsf{KAT} \models p = q \iff G(p) = G(q) \ .$$

*In other words, the equational theories of the Kleene algebras with tests and the *-continuous Kleene algebras with tests coincide.*

Proof. The forward implication is immediate, since $\mathcal{G}$ is a Kleene algebra with tests.

For the reverse implication, suppose $G(p) = G(q)$. By Lemma 10(i) and Theorem 5, $G(\widehat{p}) = G(\widehat{q})$. By Lemma 10(ii), $R(\widehat{p}) = R(\widehat{q})$. By the completeness result of [20], $\mathsf{KA} \models \widehat{p} = \widehat{q}$. Combining this with Lemma 10(i), we have $\mathsf{KAT} \models p = q$. □

Since we have shown that the equational theories of the Kleene algebras with tests and the *-continuous Kleene algebras with tests coincide, we can henceforth write $\models p = q$ unambiguously in place of $\mathsf{KAT}^* \models p = q$ or $\mathsf{KAT} \models p = q$.

8 Eliminating Hypotheses $r = 0$

Cohen [9] shows that in Kleene algebra, any equational implication of the form $r = 0 \to p = q$ reduces efficiently to a single equation. In this section we simplify Cohen's proof and extend it to handle Kleene algebras with tests.

In the proof of the main result of [21], we needed to reason in the presence of extra commutativity conditions of the form $bp = pb$, where b is a test. By Lemma 1, this equation is equivalent to $bp\bar{b} + \bar{b}pb = 0$. By the results of this section, we can eliminate conditions of this form, thus the system used in [21] reduces efficiently to the equational theory of Kleene algebra with tests. Reasoning in the presence of commutativity conditions is undecidable in general [8] (see also [21]).

Let $p, q, r \in T_{\Sigma,\mathsf{B}}$. Let u be the *universal expression* $(p_1 + \cdots + p_m)^*$, where $\Sigma = \{p_1, \ldots, p_m\}$. Under the standard interpretation over the language-theoretic model $\mathcal{G}$, the term u represents the set of all guarded strings.

The main property of the universal expression is that for any $x \in T_{\Sigma,\mathsf{B}}$, $\vDash x \leq u$. This can be shown easily in two steps: first, $\vDash x \leq x'$, where x' is obtained from x by deleting all Boolean symbols; this holds because $\vDash b \leq 1$ for all Boolean expressions b. Then, $\vDash x' \leq u$ by ordinary Kleene algebra.

Theorem 12 *The following are equivalent:*

(i) $\mathsf{KAT} \models r = 0 \to p = q$

(ii) $\mathsf{KAT}^* \models r = 0 \to p = q$

(iii) $\models p + uru = q + uru.$

Note that the equivalence of (i) and (ii) does not follow immediately from Theorem 11, since they are not equations but equational implications.

Proof. We first define a congruence on the set $T_{\Sigma,\mathsf{B}}$ of terms in the language of Kleene algebra with tests. For $s, t \in T_{\Sigma,\mathsf{B}}$, define

$$s \equiv t \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \vDash s + uru = t + uru \ .$$

The relation $\equiv$ is an equivalence relation. We show that it is a *-continuous Kleene algebra congruence.

If $s = t$ is a theorem of KAT, then $s \equiv t$, since $\vDash s = t$ implies $\vDash s + uru = t + uru$.

To show $\equiv$ is a congruence with respect to $+$, we need to show that $s \equiv t$ implies $s + w \equiv t + w$. But this says only that $\vDash s + uru = t + uru$ implies $\vDash s + w + uru = t + w + uru$, which is immediately apparent.

To show $\equiv$ is a congruence with respect to $\cdot$, we need to show that $s \equiv t$ implies $sw \equiv tw$ and $ws \equiv wt$. We establish the former; the latter follows by symmetry.

$$\begin{aligned}
&\vDash s + uru = t + uru \\
&\quad \Rightarrow \;\; \vDash (s + uru)w = (t + uru)w \\
&\quad \Rightarrow \;\; \vDash sw + uruw = tw + uruw \\
&\quad \Rightarrow \;\; \vDash sw + uruw + uru = tw + uruw + uru \\
&\quad \Rightarrow \;\; \vDash sw + uru = tw + uru \ .
\end{aligned}$$

To show $\equiv$ is a congruence with respect to *, we need to show that $s \equiv t$ implies $s^* \equiv t^*$.

$$\begin{aligned}
&\vDash s + uru = t + uru \\
&\quad \Rightarrow \;\; \vDash (s + uru)^* = (t + uru)^* \\
&\quad \Rightarrow \;\; \vDash s^*(urus^*)^* = t^*(urut^*)^* \\
&\quad \Rightarrow \;\; \vDash s^*(1 + urus^*(urus^*)^*) = t^*(1 + urut^*(urut^*)^*) \\
&\quad \Rightarrow \;\; \vDash s^* + s^*urus^*(urus^*)^* = t^* + t^*urut^*(urut^*)^* \\
&\quad \Rightarrow \;\; \vDash s^* + s^*urus^*(urus^*)^* + uru = t^* + t^*urut^*(urut^*)^* + uru \\
&\quad \Rightarrow \;\; \vDash s^* + uru = t^* + uru \ .
\end{aligned}$$

To show $\equiv$ is a congruence with respect to $\bar{\ }$, we need to show that for Boolean terms b, c, if $b \equiv c$ then $\overline{b} \equiv \overline{c}$. This case follows from previous results. If $b \equiv c$, then $b + \overline{c} \equiv c + \overline{c} \equiv 1$, thus $\overline{c}\overline{b} \equiv (b + \overline{c})\overline{b} \equiv \overline{b}$. By symmetry, $\overline{c}\overline{b} \equiv \overline{c}$, therefore $\overline{b} \equiv \overline{c}$.

Finally, we show that $\equiv$ respects the *-continuity condition (7), we only need to show (9); *i.e.*, if $st^nv + y \equiv y$ for all n, then $st^*v + y \equiv y$:

$$\begin{aligned}
&\vDash (st^nv + y) + uru = y + uru \text{ for all } n \\
&\quad \Rightarrow \;\; \vDash st^nv + (y + uru) = y + uru \text{ for all } n \\
&\quad \Rightarrow \;\; \vDash st^*v + (y + uru) = y + uru \qquad (15) \\
&\quad \Rightarrow \;\; \vDash (st^*v + y) + uru = y + uru \ .
\end{aligned}$$

The crucial step (15) follows from the fact that if $st^nv \leq y + uru$ for all n in all *-continuous Kleene algebras, then $st^*v \leq y + uru$ in all *-continuous Kleene algebras.

Since $\equiv$ is a KAT^* congruence on $T_{\Sigma,\mathsf{B}}$, we can form the quotient $T_{\Sigma,\mathsf{B}}/\equiv$ and canonical interpretation $s \mapsto [s]$, where $[s]$ denotes the $\equiv$-congruence class of s, and this structure is a *-continuous Kleene algebra with tests. The equation $r = 0$ is satisfied under this interpretation, since

$$\vDash r + uru = uru = 0 + uru \ ,$$

so $r \equiv 0$.

Now we are ready to prove the equivalence of the three conditions in the statement of the theorem.

(i) $\Rightarrow$ (ii) Any formula true in all Kleene algebras with tests is certainly true in all *-continuous Kleene algebras with tests.

(ii) $\Rightarrow$ (iii) If $\mathsf{KAT}^* \models r = 0 \rightarrow p = q$, then since $T_{\Sigma,\mathsf{B}}/\equiv$ is a *-continuous Kleene algebra with tests and $T_{\Sigma,\mathsf{B}}/\equiv, [\ ] \vDash r = 0$, we have $T_{\Sigma,\mathsf{B}}/\equiv, [\ ] \vDash p = q$. By definition, $p \equiv q$, which is what we wanted to show.

(iii) $\Rightarrow$ (i) Suppose $\models p + uru = q + uru$. Let $\mathcal{K}$ be an arbitrary Kleene algebra with tests and let I be an arbitrary interpretation over $\mathcal{K}$ such that $\mathcal{K}, I \models r = 0$. Then $\mathcal{K}, I \models p = p + uru = q + uru = q$. Since $\mathcal{K}$ and I were arbitrary, $\mathsf{KAT} \models r = 0 \rightarrow p = q$. $\square$

9 Decidability

Once we have Theorem 8, decidability of the equational theory of Kleene algebra with tests follows almost immediately from a simple reduction to Propositional Dynamic Logic (PDL). Any term in the language of KAT is a program of PDL (after replacing Boolean terms b with PDL tests $b?$), and it is known that two such terms p and q represent the same binary relation in all relational structures iff

$$\mathsf{PDL} \models \langle p \rangle c \leftrightarrow \langle q \rangle c \ ,$$

where c is a new primitive proposition symbol [13]. By Theorems 8 and 11, this is tantamount to deciding KAT-equivalence.

PDL is known to be exponential time complete [13, 27], thus the equational theory of KAT is decidable in no more than exponential time. It is at least *PSPACE*-hard, since the equational theory of Kleene algebras is [35].

It can be shown by different methods that the equational theory of KAT is *PSPACE*-complete. This argument is considerably more involved and will be the subject of a forthcoming paper.

Acknowledgements

Ernie Cohen provided valuable comments. The support of the National Science Foundation under grant CCR-9317320 is gratefully acknowledged. The second author is supported on a National Science Foundation Graduate Fellowship.

References

[1] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1975.

[2] S. Anderaa. On the algebra of regular expressions. *Appl. Math.*, pages 1–18, January 1965.

[3] K. V. Archangelsky. A new finite complete solvable quasiequational calculus for algebra of regular languages. Manuscript, Kiev State University, 1992.

[4] Roland Carl Backhouse. *Closure Algorithms and the Star-Height Problem of Regular Languages.* PhD thesis, Imperial College, 1975.

[5] Stephen L. Bloom and Zoltán Ésik. Equational axioms for regular sets. *Mathematical Structures in Computer Science*, 3:1–24, 1993.

[6] Maurice Boffa. Une remarque sur les systèmes complets d'identités rationnelles. *Informatique théoretique et Applications/Theoretical Informatics and Applications*, 24(4):419–423, 1990.

[7] Maurice Boffa. Une condition impliquant toutes les identités rationnelles. *Informatique théoretique et Applications/Theoretical Informatics and Applications*, 29(6):515–518, 1995.

[8] Ernie Cohen, February 1994. Personal communication.

[9] Ernie Cohen. Hypotheses in Kleene algebra. `ftp://ftp.bellcore.com/pub/ernie/research/homepage.html`, April 1994.

[10] Ernie Cohen. Lazy caching. `ftp://ftp.bellcore.com/pub/ernie/research/homepage.html`, 1994.

[11] Ernie Cohen. Using Kleene algebra to reason about concurrency control. `ftp://ftp.bellcore.com/pub/ernie/research/homepage.html`, 1994.

[12] John Horton Conway. *Regular Algebra and Finite Machines*. Chapman and Hall, 1971.

[13] M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *J. Comput. Syst. Sci.*, 18(2):194–211, 1979.

[14] David Harel. On folk theorems. *Comm. Assoc. Comput. Mach.*, 23(7):379–389, July 1980.

[15] Kazuo Iwano and Kenneth Steiglitz. A semiring on convex polygons and zero-sum cycle problems. *SIAM J. Comput.*, 19(5):883–901, 1990.

[16] Stephen C. Kleene. Representation of events in nerve nets and finite automata. In Shannon and McCarthy, editors, *Automata Studies*, pages 3–41. Princeton University Press, 1956.

[17] Dexter Kozen. On induction vs. *-continuity. In Kozen, editor, *Proc. Workshop on Logic of Programs*, volume 131 of *Lect. Notes in Comput. Sci.*, pages 167–176. Springer, 1981.

[18] Dexter Kozen. On Kleene algebras and closed semirings. In Rovan, editor, *Proc. Math. Found. Comput. Sci.*, volume 452 of *Lect. Notes in Comput. Sci.*, pages 26–47. Springer, 1990.

[19] Dexter Kozen. *The Design and Analysis of Algorithms*. Springer-Verlag, 1991. ISBN 0-387-97687-6. 322 pages.

[20] Dexter Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Infor. and Comput.*, 110(2):366–390, May 1994.

[21] Dexter Kozen. Kleene algebra with tests and commutativity conditions. In T. Margaria and B. Steffen, editors, *Proc. Second Int. Workshop Tools and Algorithms for the Construction and Analysis of Systems (TACAS'96)*, volume 1055 of *Lect. Notes in Comput. Sci.*, pages 14–33. Springer, March 1996.

[22] Daniel Krob. A complete system of B-rational identities. *Theoretical Computer Science*, 89(2):207–343, October 1991.

[23] Werner Kuich. The Kleene and Parikh theorem in complete semirings. In Ottmann, editor, *Proc. 14th Colloq. Automata, Languages, and Programming*, volume 267 of *Lect. Notes in Comput. Sci.*, pages 212–225. EATCS, Springer, 1987.

[24] Werner Kuich and Arto Salomaa. *Semirings, Automata, and Languages*. Springer, 1986.

[25] G. Mirkowska. *Algorithmic Logic and its Applications*. PhD thesis, University of Warsaw, 1972. in Polish.

[26] K. C. Ng. *Relation Algebras with Transitive Closure*. PhD thesis, University of California, Berkeley, 1984.

[27] V. R. Pratt. Models of program logics. In *Proc. 20th Symp. Found. Comput. Sci.*, pages 115–122. IEEE, 1979.

[28] V. R. Pratt. Dynamic algebras and the nature of induction. In *Proc. 12th Symp. Theory of Comput.*, pages 22–28. ACM, 1980.

[29] V. R. Pratt. A near-optimal method for reasoning about actions. *J. Comput. Syst. Sci.*, 20(2):231–254, 1980.

[30] Vaughan Pratt. Dynamic algebras as a well-behaved fragment of relation algebras. In D. Pigozzi, editor, *Proc. Conf. on Algebra and Computer Science*, volume 425 of *Lect. Notes in Comput. Sci.*, pages 77–110. Springer, June 1988.

[31] Vaughan Pratt. Action logic and pure induction. In J. van Eijck, editor, *Proc. Logics in AI: European Workshop JELIA '90*, volume 478 of *Lect. Notes in Comput. Sci.*, pages 97–120. Springer, September 1990.

[32] V. N. Redko. On defining relations for the algebra of regular events. *Ukrain. Mat. Z.*, 16:120–126, 1964. In Russian.

[33] Jacques Sakarovitch. Kleene's Theorem revisited: a formal path from Kleene to Chomsky. In A. Kelemenova and J. Keleman, editors, *Trends, Techniques, and Problems in Theoretical Computer Science*, volume 281 of *Lect. Notes in Computer Science*, pages 39–50. Springer, 1987.

[34] Arto Salomaa. Two complete axiom systems for the algebra of regular events. *J. Assoc. Comput. Mach.*, 13(1):158–169, January 1966.

[35] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time. In *Proc. 5th Symp. Theory of Computing*, pages 1–9. ACM, 1973.

[36] A. Tarski. On the calculus of relations. *J. Symb. Logic*, 6(3):65–106, 1941.