On the Elimination of Hypotheses in Kleene Algebra with Tests

Chris Hardin
Department of Mathematics
Cornell University
Ithaca, New York 14853-4201, USA
hardin@math.cornell.edu

Dexter Kozen
Department of Computer Science
Cornell University
Ithaca, New York 14853-7501, USA
kozen@cs.cornell.edu

November 18, 2002

Abstract

The validity problem for certain universal Horn formulas of Kleene algebra with tests (KAT) can be efficiently reduced to the equational theory. This reduction is known as *elimination of hypotheses*. Hypotheses are used to describe the interaction of atomic programs and tests and are an essential component of practical program verification with KAT. The ability to eliminate hypotheses of a certain form means that the Horn theory with premises of that form remains decidable in *PSPACE*. It was known (Cohen 1994, Kozen and Smith 1996, Kozen 1997) how to eliminate hypotheses of the form $q = 0$. In this paper we show how to eliminate hypotheses of the form $cp = c$ for atomic p. Hypotheses of this form are useful in eliminating redundant code and arise quite often in the verification of compiler optimizations (Kozen and Patron 2000).

1 Introduction

Kleene algebra with tests (KAT), introduced in [12], is an equational system for program verification that combines Kleene algebra (KA) with Boolean algebra. KAT has been applied successfully in various low-level verification tasks involving communication protocols, basic safety analysis, source-to-source program transformation, concurrency control, and compiler optimization [3, 4, 5, 12, 15, 1, 2]. The system subsumes Hoare logic and is deductively complete for partial correctness over relational models [14].

A useful feature of KAT in practical verification tasks is its ability to accommodate basic equational hypotheses regarding the interaction of atomic instructions and tests. This feature makes KAT ideal for static analysis of complicated code fragments based on the behavior of their atomic parts.

For example, consider the case of an assertion b that holds at some point in a program immediately before an action p, and suppose we know that the execution of p cannot affect the truth of b. For instance, p might be an assignment such as $x := 3$ and b might be a test such as $y = 4$ that refers to a different variable. In KAT, the independence of p and b is modeled by a commutativity condition $pb = bp$, which is typically postulated as an assumption. The rules of equational logic allow pb to be substituted for bp and vice-versa; intuitively, if p and b are adjacent in the program, they can exchange positions.

Similarly, assertions arising from the execution of actions can be introduced and eliminated as needed using equational assumptions of the form $p = pc$. For example, if p is the assignment $x := 3$ and c is the assertion $x = 3$, then any execution of p causes c to hold immediately afterward. Using $p = pc$, one can introduce the assertion c immediately following any occurrence of p in the program, then move it around using commutativity conditions as described in the preceding paragraph. If an occurrence of c can be moved to a position immediately preceding some other occurrence of p, then that occurrence of p can be eliminated, since it is redundant: if x already has the value 3, there is no need to assign 3 to it again. Formally, we postulate $cp = c$. This technique is useful in the verification of various compiler optimizations that eliminate unnecessary code, such as the loading of a register with a constant value inside a loop. See [15] for many examples of this type.

In such proofs, the underlying first-order semantics of p and c (i.e., that p is $x := 3$ and c is $x = 3$) are used to establish the correctness of the premises $p = pc$ and $cp = c$; but once this is done, the argument reverts to purely propositional reasoning, using $p = pc$ and $cp = c$ as equational assumptions without reference to their semantics.

Much attention has focused on the equational theory of KA and KAT. The axioms of KAT are known to be deductively complete for the equational theory of language and relational models, and validity is decidable in *PSPACE* [16, 6]. But because of the practical importance of premises, it is the universal Horn theory that is of more interest; that is, the set of valid sentences of the form

$$p_1 = q_1 \wedge \cdots \wedge p_n = q_n \quad \to \quad p = q, \tag{1}$$

where the atomic symbols are implicitly universally quantified. Typically, the premises $p_i = q_i$ are assumptions such as $bp = pb$, $p = pc$, and $cp = c$ regarding the interaction of atomic programs and tests, and the conclusion $p = q$ represents the equivalence of the optimized and unoptimized program. The necessary premises are obtained by inspection of the program and their validity may depend on properties of the domain of computation, but they are usually quite simple and easy to verify by inspection, since they typically only involve atomic programs and tests. Once the premises are established, the proof of (1) is purely propositional. This ability to introduce premises as needed is one of the features that makes KAT so versatile. By comparison, Hoare logic has only the assignment rule, which is much more limited. In addition, this style of reasoning allows a clean separation between first-order interpreted reasoning to justify the premises $p_1 = q_1 \wedge \cdots \wedge p_n = q_n$ and purely propositional reasoning to establish that the conclusion $p = q$ follows from the premises.

Unfortunately, the Horn theory is computationally more complex than the equational theory. The general Horn theory for *-continuous algebras is Π^1_1-complete. Even when the premises are restricted to commutativity conditions of the form $pq = qp$ for atomic actions p and q, the validity problem is Π^0_1-complete [13].

However, sometimes the validity of universal Horn formulas with premises of a certain restricted form can be efficiently reduced to the equational theory. This reduction is known as *elimination of hypotheses*. Cohen [3] was the first to identify this as an important issue. He showed how to eliminate hypotheses of the form $q = 0$ in KA; thus the Horn theory of KA with premises of this form remains decidable in *PSPACE*. These results were generalized to KAT in [16]. This is good news for many of the program verification tasks mentioned above, since in many cases the premises are of this form. For example, the commutativity condition $bp = pb$ is equivalent to the condition $bp\overline{b} + \overline{b}pb = 0$, and the condition $pc = p$ is equivalent to the condition $p\overline{c} = 0$. All partial correctness assertions of Hoare logic are of this form as well: the Hoare partial correctness assertion $\{b\}\ p\ \{c\}$ is equivalent to the equation $bp\overline{c} = 0$. For this reason, we call Horn formulas with premises of the restricted form $q = 0$ *Hoare formulas*.

The general question thus arises: under what conditions can hypotheses can be eliminated? In other words, under what restrictions on the premises does the validity of Horn formulas reduce to the validity of equations? Although we do not have a general answer to this question, we can extend the class of useful premises for which elimination is possible: we show in this paper how to eliminate hypotheses of the form $cp = c$ for atomic p. Equations of this form are not equivalent to equations of the form $q = 0$ in general.

Before we go further, there are several subtleties in the question itself that must be addressed. One issue is that unlike the equational theory, the question depends on the class of models under consideration. In order of increasing restriction, one might consider validity over unrestricted (KAT), *-continuous (KAT^*), or relational (REL) Kleene algebras with tests. The equational theories of all these classes coincide [16], but this is not true of their Horn theories. The Horn theories of KAT and KAT^* must differ, since the former is recursively enumerable—it is defined by a finite quasiequational axiomatization—whereas the latter is Π^1_1-complete [13]; and the Horn theories of KAT^* and REL differ, since $p \leq 1 \to p^2 = p$ is valid in all relational models, but not in all *-continuous KATs; for example, not in the min,+ algebra.

The results of [3, 16] on the elimination of hypotheses of the form $q = 0$ were initially shown to hold for *-continuous and general KA and KAT, but the corresponding result for relational models does not follow from these results or their proofs. This was a subtle but crucial oversight, since in programming language semantics, it is the relational models that are of primary interest. The situation was rectified in [14], where it was established that the Hoare theories of KAT, KAT^*, and REL coincide, and that the same reduction also works for relational models.

Cohen [3] shows also that hypotheses of the form $p \leq 1$ can be eliminated, provided p contains no occurrence of a composition operator. However, this result is more problematic. His reduction is valid when interpreted over the classes of all Kleene algebras or all *-continuous Kleene algebras; however, it fails when restricted to relational models. In fact, an example formula on which it fails is the formula $p \leq 1 \rightarrow p^2 = p$ mentioned above. Since the reduction does not work for relational models, and since it is the relational models that are of primary interest in program semantics, the situation is not completely satisfactory.

Another issue is that one would like to eliminate hypotheses of the form $p \leq 1$ or $q = 0$ simultaneously. Cohen does not address this issue. In the case of premises of the form $q = 0$, it is easy to see how to combine several of them into one: the conjunction $q_1 = 0 \wedge \cdots \wedge q_n = 0$ is equivalent to the single equation $q_1 + \cdots + q_n = 0$. A similar construction can be used to combine several premises of the form $p \leq 1$ into one. However, it is not immediately clear how to handle both forms simultaneously.

In this paper we consider premises of the form $cp = c$ for atomic p. The utility of such premises in practical verification has been argued above. Such equations are not equivalent to any equation of the form $q = 0$, and the construction we use is quite different. We show that an arbitrary finite set of premises of this form in conjunction with arbitrarily many premises of the form $q = 0$ can be simultaneously eliminated, giving an efficient reduction of the Horn theory with premises of the form $cp = c$ for atomic p or $q = 0$ to the equational theory. Moreover, this result holds irrespective of whether the class of interpretations is KAT, KAT^*, or REL; that is, the Horn theories of these three classes of models, restricted to premises of the form $cp = c$ for atomic p or $q = 0$, coincide. Thus the Horn theory with premises of this form remains decidable in *PSPACE*.

2 Preliminary Definitions

2.1 Kleene Algebra

Kleene algebra (KA) is the algebra of regular expressions [10, 7]. The axiomatization used here is from [11]. A *Kleene algebra* is an algebraic structure $(K, +, \cdot, ^*, 0, 1)$ that is an idempotent semiring under $+, \cdot, 0, 1$ such that p^*q is the $\leq$-least solution to $q+px \leq x$ and qp^* is the $\leq$-least solution to $q+xp \leq x$, where $\leq$ refers to the natural partial order on K: $p \leq q \stackrel{\text{def}}{\Longleftrightarrow} p + q = q$. This is a universal Horn axiomatization. A Kleene algebra is *-*continuous* if it satisfies the stronger infinitary property $pq^*r = \sup_n pq^nr$.

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

Kleene algebra is a versatile system with many useful interpretations. Standard 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. Other more unusual interpretations include the min,+ algebra, also known as the *tropical semiring*, used in shortest path algorithms, and models consisting of convex polyhedra used in computational geometry.

The completeness result of [11] says that all true identities between regular expressions interpreted as regular sets of strings are derivable from the axioms of Kleene algebra. In other words, the algebra of regular sets of strings over the finite alphabet P is the free Kleene algebra on generators P. The axioms are also complete for the equational theory of relational models.

See [11] for a more thorough introduction.

2.2 Kleene Algebra with Tests

Kleene algebras with tests (KAT) were introduced in [12]. We give a brief introduction here, but refer the reader to [12, 14, 17] for a more detailed treatment.

A Kleene algebra with tests is just a Kleene algebra with an embedded Boolean subalgebra. That is, it is a two-sorted structure

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

such that

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

The Boolean complementation operator $^-$ is defined only on B. Elements of B are called *tests*. The letters $p, q, r, s, \ldots$ denote arbitrary elements of K and $a, b, c, \ldots$ denote tests.

The encoding of the **while** program constructs is as in propositional Dynamic Logic [8]:

$$\begin{aligned} p\ ;\ q &\stackrel{\text{def}}{=} pq \\ \textbf{if } b \textbf{ then } p \textbf{ else } q &\stackrel{\text{def}}{=} bp + \overline{b}q \\ \textbf{while } b \textbf{ do } p &\stackrel{\text{def}}{=} (bp)^*\overline{b}. \end{aligned}$$

The Hoare partial correctness assertion $\{b\}\,p\,\{c\}$ is expressed as an equation $bp\overline{c} = 0$, or equivalently, $bp = bpc$. All Hoare rules are derivable in KAT; indeed, KAT is deductively complete for relationally valid propositional Hoare-style rules involving partial correctness assertions [14] (propositional Hoare logic is not).

Let P and B be disjoint sets of symbols called the *atomic actions* and *atomic tests*, respectively. We denote by $\mathsf{RExp}_{\mathsf{P},\mathsf{B}}$ the set of terms of the language of KAT over P and B. A *test* over B is just a Boolean combination of elements of B. The set of tests over B is denoted $\mathsf{Bool}_{\mathsf{B}}$.

Lemma 2.1 *The following are equivalent in* KAT*:*

(i) $cp = c$

(ii) $cp + \overline{c} = 1$

(iii) $p = \overline{c}p + c$.

Proof. For (i) → (ii), replace cp by c on the left-hand side of (ii) and use the Boolean algebra axiom $c+\overline{c} = 1$. For (i) → (iii), replace c by cp on the right-hand side of (iii) and use distributivity and the Boolean algebra axiom $c + \overline{c} = 1$. For (ii) → (i) and (iii) → (i), multiply both sides of (ii) or (iii) on the left by c and use distributivity and the Boolean algebra axioms $c\overline{c} = 0$ and $cc = c$. □

We write $\mathsf{KAT} \vDash \varphi$ (respectively, $\mathsf{KAT}^* \vDash \varphi$) if φ holds under all interpretations over Kleene algebras with tests (respectively, *-continuous Kleene algebras with tests).

2.3 Kripke Frames

For applications in program verification, we usually interpret programs and tests either as sets of traces or as binary relations on a set of states. Both these classes of algebras are defined in terms of *Kripke frames*. A Kripke frame over a set of atomic programs P and a set of atomic tests B is a structure $(K,\ \mathfrak{m}_K)$, where K is a set of *states*, $\mathfrak{m}_K : \mathsf{P} \to 2^{K\times K}$, and $\mathfrak{m}_K : \mathsf{B} \to 2^K$.

2.4 Relational Models

The set of all binary relations on a Kripke frame K forms a KAT under the standard binary relation-theoretic interpretation of the KAT operators. The operator $\cdot$ is interpreted as relational composition, $+$ as union, 0 and 1 as the empty relation and the identity relation on K, respectively, and * as reflexive transitive closure. The Boolean elements are subsets of the identity relation. One can define a canonical interpretation $[\ ]_K$: $\mathsf{RExp}_{\mathsf{P,B}} \to 2^{K\times K}$ by

$$[p]_K \stackrel{\text{def}}{=} \mathfrak{m}_K(p), \quad p \in \mathsf{P} \qquad\qquad [b]_K \stackrel{\text{def}}{=} \{(u,u) \mid u \in \mathfrak{m}_K(b)\}, \quad b \in \mathsf{B},$$

extended homomorphically. A binary relation is *regular* if it is $[p]_K$ for some $p \in \mathsf{RExp}_{\mathsf{P,B}}$. The relational algebra consisting of all regular binary relations on K is denoted Rel_K.

We write $\mathsf{Rel}_K \vDash \varphi$ if the formula φ is true in this model under the canonical interpretation $[\ ]_K$, and we write $\mathsf{REL} \vDash \varphi$ if φ is true under all such interpretations. If φ is a single equation, we can omit KAT, KAT*, or REL before the symbol $\vDash$, since these classes of algebras are known to have the same equational theory [16].

2.5 Trace Models

A *trace* in a Kripke frame K is a sequence $u_0p_0u_1 \cdots u_{n-1}p_{n-1}u_n$, where $n \geq 0$, $u_i \in K$, $p_i \in \mathsf{P}$, and $(u_i, u_{i+1}) \in \mathfrak{m}_K(p_i)$ for $0 \leq i \leq n-1$. The set of all traces in K is denoted Traces_K. We denote traces by $\sigma, \tau, \ldots$. The first and last states of a trace σ are denoted $\mathbf{first}(\sigma)$ and $\mathbf{last}(\sigma)$, respectively. If $\mathbf{last}(\sigma) = \mathbf{first}(\tau)$, we can fuse σ and τ to get the trace $\sigma\tau$.

The powerset of Traces_K forms a KAT in which $+$ is interpreted as set union, $\cdot$ as the operation

$$AB \quad \stackrel{\text{def}}{=} \quad \{\sigma\tau \mid \sigma \in A,\ \tau \in B,\ \mathbf{last}(\sigma) = \mathbf{first}(\tau)\},$$

0 and 1 as $\varnothing$ and K, respectively, and A^* as the union of all finite powers of A. The Boolean elements are the subsets of K, the sets of traces of length 0. A canonical interpretation $[\![\ ]\!]_K$ for KAT expressions over P and B is given by

$$[\![p]\!]_K \stackrel{\text{def}}{=} \{upv \mid (u,v) \in \mathfrak{m}_K(p)\}, \quad p \in \mathsf{P} \qquad\qquad [\![b]\!]_K \stackrel{\text{def}}{=} \mathfrak{m}_K(b), \quad b \in \mathsf{B},$$

extended homomorphically. A set of traces is *regular* if it is $[\![p]\!]_K$ for some KAT expression p. The subalgebra of all regular sets of traces of K is denoted Tr_K.

A homomorphism involving trace or relation algebras on Kripke frames over P, B is *canonical* if it commutes with the canonical interpretations $[\![\ ]\!]_K$ or $[\ ]_K$. For example, the map $\mathsf{Ext}(A) = \{(\mathbf{first}(\sigma), \mathbf{last}(\sigma)) \mid \sigma \in A\}$ is a canonical homomorphism $\mathsf{Tr}_K \to \mathsf{Rel}_K$, since $\mathsf{Ext}([\![p]\!]_K) = [p]_K$ for all $p \in \mathsf{RExp}_{\mathsf{P,B}}$.

2.6 Guarded Strings

When B is finite, a language-theoretic interpretation is given by the algebra of regular sets of *guarded strings* [9, 16]. Let $\mathsf{Atoms}_\mathsf{B}$ denote the set of atoms (minimal nonzero elements) of the free Boolean algebra generated by B. We use the symbols $\alpha, \beta, \ldots$ exclusively for atoms. For an atom α and a test b, we write $\alpha \leq b$ if $\alpha \to b$ is a propositional tautology.

A *guarded string* over P, B is a trace in the Kripke frame G whose states are $\mathsf{Atoms_B}$ and

$$\begin{aligned} \mathfrak{m}_G(p) &\stackrel{\text{def}}{=} \mathsf{Atoms_B} \times \mathsf{Atoms_B}, \quad p \in \mathsf{P} \\ \mathfrak{m}_G(b) &\stackrel{\text{def}}{=} \{\alpha \in \mathsf{Atoms_B} \mid \alpha \leq b\}, \quad b \in \mathsf{B}. \end{aligned}$$

Thus a guarded string is just a sequence $\alpha_0 p_0 \alpha_1 \cdots \alpha_{n-1} p_{n-1} \alpha_n$, where the $\alpha_i \in \mathsf{Atoms_B}$ and $p_i \in \mathsf{P}$, and Traces_G is the set of all guarded strings over P, B. Each KAT term $p \in \mathsf{RExp_{P,B}}$ denotes a set $[\![p]\!]_G$ of guarded strings under the canonical interpretation defined in Section 2.5. A guarded string σ is itself a member of $\mathsf{RExp_{P,B}}$, and $[\![\sigma]\!]_G = \{\sigma\}$.

The trace algebra Tr_G of regular sets of guarded strings over P, B forms the free Kleene algebra with tests on generators P, B; in other words, $[\![p]\!]_G = [\![q]\!]_G$ iff $p = q$ is a theorem of KAT [16].

3 Main Results

In this section we show how to eliminate hypotheses of the form $cp = c$ for atomic p. Before we do this, we argue that this result does not follow from any previously known results on the elimination of hypotheses.

Theorem 3.1 *Let p be an atomic action and c a test that does not vanish tautologically. The equation $cp = c$ is not equivalent to any inequality of the form $x \leq a$ for a test a. In particular, $cp = c$ is not equivalent to $x \leq 1$ or $x = 0$. Moreover, this holds even restricted to relational models.*

Proof. Let a be a test. Suppose for a contradiction that

$$\mathsf{REL} \models cp = c \leftrightarrow x \leq a. \tag{2}$$

Let P and B be the sets of all atomic actions and tests, respectively, occurring in (2). Let u be the universal expression $(\sum_{q \in \mathsf{P}} q)^*$. We claim first that

$$\models x \leq uc(\sum_{b \in \mathsf{B}} bp\bar{b} + \bar{b}pb)u + aua. \tag{3}$$

Let $[\![\ ]\!]_G$ be the canonical interpretation $\mathsf{RExp_{P,B}} \to \mathsf{Tr}_G$. Let

$$\sigma = \alpha_0 p_0 \alpha_1 \cdots \alpha_{n-1} p_{n-1} \alpha_n$$

be an arbitrary guarded string in $[\![x]\!]_G$. Suppose that

$$\sigma \quad \notin \quad [\![uc(\textstyle\sum_{b\in\mathsf{B}} bp\overline{b} + \overline{b}pb)u]\!]_G. \tag{4}$$

Then for all i in the range $0 \le i \le n-1$, if $p_i = p$ and $\alpha_i \le c$, then $\alpha_i = \alpha_{i+1}$. Let $(K,\ \mathfrak{m}_K)$ be a Kripke frame with

$$\begin{aligned}
K &\stackrel{\text{def}}{=} \mathsf{Atoms}_\mathsf{B}, \\
\mathfrak{m}_K(b) &\stackrel{\text{def}}{=} \{\alpha \mid \alpha \le b\}, \quad b \in \mathsf{B}, \\
\mathfrak{m}_K(p) &\stackrel{\text{def}}{=} \{(\alpha,\alpha) \mid \alpha \le c\} \cup \{(\alpha,\beta) \mid \alpha \le \overline{c},\ \beta \in \mathsf{Atoms}_\mathsf{B}\} \\
\mathfrak{m}_K(q) &\stackrel{\text{def}}{=} \{(\alpha,\beta) \mid \alpha,\beta \in \mathsf{Atoms}_\mathsf{B}\}, \quad q \in \mathsf{P},\ q \ne p.
\end{aligned}$$

In this Kripke frame, for any i,

- if $p_i = p$ and $\alpha_i \le c$, then $\alpha_i = \alpha_{i+1}$, therefore $(\alpha_i, \alpha_{i+1}) \in [p_i]_K$;
- if $p_i = p$ and $\alpha_i \le \overline{c}$, or if $p_i \ne p$, then $(\alpha_i, \alpha_{i+1}) \in [p_i]_K$.

Thus in any case, $(\alpha_i, \alpha_{i+1}) \in [p_i]_K$. Moreover, $[\alpha_i]_K = \{(\alpha_i, \alpha_i)\}$. Thus

$$\begin{aligned}
[\sigma]_K &= [\alpha_0]_K \circ [p_0]_K \circ [\alpha_1]_K \circ \cdots \circ [\alpha_{n-1}]_K \circ [p_{n-1}]_K \circ [\alpha_n]_K \\
&= \{(\alpha_0, \alpha_n)\}.
\end{aligned}$$

Also, $[c]_K = \{(\alpha,\alpha) \mid \alpha \in \mathfrak{m}_K(c)\} = \{(\alpha,\alpha) \mid \alpha \le c\}$ and $[p]_K = \mathfrak{m}_K(p)$, therefore

$$\begin{aligned}
[cp]_K &= [c]_K \circ [p]_K \\
&= \{(\alpha,\alpha) \mid \alpha \le c\} \circ (\{(\alpha,\alpha) \mid \alpha \le c\} \cup \{(\alpha,\beta) \mid \alpha \le \overline{c},\ \beta \in \mathsf{Atoms}_\mathsf{B}\}) \\
&= \{(\alpha,\alpha) \mid \alpha \le c\} \\
&= [c]_K,
\end{aligned}$$

thus $\mathsf{REL}_K \vDash cp = c$. Using (2) in the direction $\rightarrow$, we have $\mathsf{REL}_K \vDash x \le a$. Since $\sigma \le x$, $\mathsf{REL}_K \vDash \sigma \le a$ as well, thus $[\sigma]_K = \{(\alpha_0, \alpha_n)\} \subseteq [a]_K = \{(\alpha,\alpha) \mid \alpha \le a\}$, therefore $\alpha_0 = \alpha_n$ and $\alpha_0 \le a$. This says that

$$\sigma \quad \in \quad [\![\textstyle\sum_{\alpha \le a} \alpha u \alpha]\!]_G \quad \subseteq \quad [\![aua]\!]_G. \tag{5}$$

We have derived (5) under the assumption (4) for arbitrary $\sigma \in [\![x]\!]_G$, thus

$$[\![x]\!]_G \quad \subseteq \quad [\![uc(\textstyle\sum_{b\in\mathsf{B}} bp\overline{b} + \overline{b}pb)u + aua]\!]_G.$$

By the completeness of KAT over the guarded string model [16], we have (3).

Now it follows from (3) that

$$\models \quad uc(\sum_{b \in \mathsf{B}} bp\overline{b} + \overline{b}pb)u + aua \ \leq \ a \quad \to \quad x \ \leq \ a,$$

and combining this with (2) in the direction $\leftarrow$, we have

$$\mathsf{REL} \quad \models \quad uc(\sum_{b \in \mathsf{B}} bp\overline{b} + \overline{b}pb)u + aua \ \leq \ a \quad \to \quad cp = c.$$

But then this should hold even under interpretations that assign 0 to each atomic action, thus

$$\mathsf{REL} \quad \models \quad 0 + a \leq a \quad \to \quad 0 = c,$$

which implies that $\models 0 = c$, contradicting the assumption that c is not tautologically false.

$\square$

The following is our main theorem.

Theorem 3.2 *Let* $s_1, \ldots, s_m \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$*,* $c_1, \ldots, c_n \in \mathsf{Bool}_\mathsf{B}$*,* $r_1, \ldots, r_n \in \mathsf{P} \cup \mathsf{Bool}_\mathsf{B}$*, and* $p, q \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$*. There exist* $\widehat{p}, \widehat{q} \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$ *such that the following are equivalent:*

(i) $\mathsf{KAT} \ \models \ \bigwedge_{i=0}^m s_i = 0 \ \wedge \ \bigwedge_{i=0}^n c_i r_i = c_i \ \to \ p = q$

(ii) $\mathsf{KAT}^* \ \models \ \bigwedge_{i=0}^m s_i = 0 \ \wedge \ \bigwedge_{i=0}^n c_i r_i = c_i \ \to \ p = q$

(iii) $\mathsf{REL} \ \models \ \bigwedge_{i=0}^m s_i = 0 \ \wedge \ \bigwedge_{i=0}^n c_i r_i = c_i \ \to \ p = q$

(iv) $\models \ \widehat{p} = \widehat{q}$.

Furthermore, $\widehat{p}$ *and* $\widehat{q}$ *can be calculated from* $s_1, \ldots, s_m$*,* $c_1, \ldots, c_n$*,* $r_1, \ldots, r_n$*,* p*, and* q *in* $PTIME$*, and any one of* (i)–(iv) *can be decided in PSPACE.*

The remainder of this paper is devoted to the proof of Theorem 3.2. First we make some simplifications.

As noted above, the conjunction $s_1 = 0 \wedge \cdots \wedge s_m = 0$ is equivalent to the single equation $s_1 + \cdots + s_m = 0$. Thus we can assume without loss of generality that $m = 1$.

We can also assume that all the r_i are in P, since if r_i is a test, we can replace the premise $c_i r_i = c_i$ with the equivalent premise $c_i \overline{r}_i = 0$, which we can handle along with the other premises $s_i = 0$.

Finally, we can assume without loss of generality that the r_i are distinct. For $c, d \in \mathsf{Bool}_\mathsf{B}$ and $r \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$, we claim that

$$\models \quad cr = c \wedge dr = d \ \leftrightarrow \ (c + d)r = c + d.$$

If $(c+d)r = c+d$, then multiplying both sides on the left by c and using Boolean algebra, we get $cr = c$. We can obtain $dr = d$ similarly. Conversely, if $cr = c$ and $dr = d$, then $(c+d)r = cr + dr = c + d$. Thus, whenever $r_i = r_j$ with $i \neq j$, we can replace the hypotheses $c_i r_i = c_i$ and $c_j r_j = c_j$ with the single equivalent hypothesis $(c_i + c_j)r_i = (c_i + c_j)$, repeating as necessary until all the r_i are distinct.

Henceforth, we fix the c_i and r_i, fix $s = s_1$, and make the additional assumptions that $m = 1$ and the r_i are all in P and distinct. As argued above, these assumptions are without loss of generality. Our proof for this special case constructs a relational model whose states are certain guarded strings, but we develop some theory first.

For $t, e_1, \ldots, e_k \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$ and $p_1, \ldots, p_k \in \mathsf{P}$, let $t[p_1/e_1, \ldots, p_k/e_k]$ denote the result of simultaneously substituting e_i for each occurrence of p_i in t, $1 \leq i \leq k$. We are particularly interested in the substitution

$$H(t) \ \stackrel{\text{def}}{=}\ t[r_1/\overline{c}_1 r_1 + c_1, \ldots, r_n/\overline{c}_n r_n + c_n].$$

The substitutions can be performed simultaneously or sequentially, and the order does not matter, since r_i does not occur in $\overline{c}_j r_j + c_j$ for $i \neq j$. This particular substitution is of interest because $c_i r_i = c_i$ is KAT-equivalent to $r_i = \overline{c}_i r_i + c_i$, as shown in Lemma 2.1.

Another vital fact is that performing the substitution H once is equivalent to performing it any number of times; that is, $\vDash H(H(t)) = H(t)$. To see this, observe that

$$(\overline{c}_i r_i + c_i)[r_i/\overline{c}_i r_i + c_i] \ = \ \overline{c}_i(\overline{c}_i r_i + c_i) + c_i \ = \ \overline{c}_i\overline{c}_i r_i + \overline{c}_i c_i + c_i \ = \ \overline{c}_i r_i + c_i.$$

The map H is a syntactic homomorphism $\mathsf{RExp}_{\mathsf{P},\mathsf{B}} \to \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$. We now indicate how this homomorphism is reflected semantically in trace models. For this purpose, we define a rewrite relation $\rhd$ on traces of a Kripke frame $(K,\ \mathfrak{m}_K)$. The relation $\rhd$ consists of n rules

$$s r_i s \rhd s \quad \text{provided} \quad s \in [\![\, c_i \,]\!]_K,$$

one rule for each $1 \leq i \leq n$. These rules may be applied to any subtrace of a trace. Thus any trace $\sigma r_i \tau$ can be rewritten to $\sigma\tau$ whenever $\mathbf{last}(\sigma) = \mathbf{first}(\tau) \in [\![\, c_i \,]\!]_K$. Every $\rhd$-reduction yields a shorter trace, and $\rhd$ is easily seen to be Church-Rosser, so every trace σ has a unique $\rhd$-normal form, which we denote by $N_K(\sigma)$. If X is a set of traces of K, let $N_K(X) \stackrel{\text{def}}{=} \{N_K(\sigma) \mid \sigma \in X\}$. Note that $N_K(N_K(\sigma)) = N_K(\sigma)$ and $N_K(\sigma\tau) = N_K(\sigma)N_K(\tau)$. Also,

$$\begin{aligned} N_K(XY) &= \{N_K(\sigma\tau) \mid \sigma \in X,\ \tau \in Y\} \\ &= \{N_K(\sigma)N_K(\tau) \mid \sigma \in X,\ \tau \in Y\} \\ &= N_K(X)N_K(Y). \end{aligned}$$

Let u be the universal term $u = (\sum_{q\in \mathsf{P}} q)^*$. Then $[\![u]\!]_K = \mathsf{Traces}_K$. Define

$$C \stackrel{\text{def}}{=} [\![u(\textstyle\sum_i c_i r_i)u)]\!]_K, \tag{6}$$

the set of all traces of the form $\cdots sr_i \cdots$ with $s \in [\![c_i]\!]_K$ for some i. Note that $\sigma\tau \in C$ iff $\sigma \in C$ or $\tau \in C$. For $X \subseteq \mathsf{Traces}_K$, define $h(X) \stackrel{\text{def}}{=} N_K(X) - C$.

Lemma 3.3 *The set $\{N_K(X) - C \mid X \subseteq \mathsf{Traces}_K\}$ is a Kleene algebra with tests under the usual interpretation of the operators on sets of traces, and h is a* KAT *homomorphism. Moreover, for all $t \in \mathsf{RExp}_{\mathsf{P,B}}$, $[\![H(t)]\!]_K = h([\![t]\!]_K)$; in other words, the following diagram commutes:*

$$\begin{tikzcd}
\mathsf{RExp}_{\mathsf{P,B}} \arrow[r, "{[\![\ ]\!]_K}"] \arrow[d, "H"'] & \mathsf{Tr}_K \arrow[d, "h"] \\
\mathsf{RExp}_{\mathsf{P,B}} \arrow[r, "{[\![\ ]\!]_K}"'] & \mathsf{Tr}_K
\end{tikzcd}$$

Proof. It is easily checked that the family of sets of the form $N_K(X) - C$ for $X \subseteq \mathsf{Traces}_K$ is closed under the usual KAT operations on sets of traces and that $h : X \mapsto N_K(X) - C$ is a homomorphism. Specifically, for any sets X, Y, X_i of traces in K and any set $B \subseteq K$,

$$\begin{aligned}
N_K(\textstyle\bigcup_i X_i) - C &= \textstyle\bigcup_i (N_K(X_i) - C) \\
N_K(XY) - C &= (N_K(X) - C)(N_K(Y) - C) \\
N_K(X^*) - C &= (N_K(X) - C)^* \\
N_K(\varnothing) - C &= \varnothing \\
N_K(K) - C &= K \\
N_K(K - B) - C &= K - (N_K(B) - C).
\end{aligned}$$

To show that $[\![H(t)]\!]_K = h([\![t]\!]_K)$ for all $t \in \mathsf{RExp}_{\mathsf{P,B}}$, since all maps in question are homomorphisms, it is enough to show it for atomic p and b. For r_i,

$$\begin{aligned}
[\![H(r_i)]\!]_K &= [\![\overline{c}_i r_i + c_i]\!]_K \\
&= [\![\overline{c}_i r_i]\!]_K \cup [\![c_i]\!]_K \\
&= \{sr_i v \mid s \in [\![\overline{c}_i]\!]_K\} \cup \{s \mid s \in [\![c_i]\!]_K\} \\
&= \{N_K(sr_i v) \mid N_K(sr_i v) \notin C\} \\
&= N_K([\![r_i]\!]_K) - C.
\end{aligned}$$

For $p \neq r_i$ for any i, since elements of $[\![p]\!]_K$ have no $\rhd$ redexes,

$$[\![H(p)]\!]_K \;=\; [\![p]\!]_K \;=\; N_K([\![p]\!]_K) \;=\; N_K([\![p]\!]_K) - C.$$

The case for tests is similar, since traces of length 0 are single states, therefore have no $\rhd$-redexes. $\square$

Lemma 3.4 *Let $r_1, \ldots, r_n$ be distinct elements of* P, $c_1, \ldots, c_n$ *tests, and* $s, p, q \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$. *The following are equivalent:*

(i) $\mathsf{KAT} \vDash s = 0 \wedge \bigwedge_{i=1}^{n} c_i r_i = c_i \ \rightarrow \ p = q$

(ii) $\mathsf{KAT}^* \vDash s = 0 \wedge \bigwedge_{i=1}^{n} c_i r_i = c_i \ \rightarrow \ p = q$

(iii) $\mathsf{REL} \vDash s = 0 \wedge \bigwedge_{i=1}^{n} c_i r_i = c_i \ \rightarrow \ p = q$

(iv) $\vDash H(p + usu) = H(q + usu)$.

Proof. The implications (i) $\Rightarrow$ (ii) $\Rightarrow$ (iii) are trivial, since $\mathsf{REL} \subseteq \mathsf{KAT}^* \subseteq \mathsf{KAT}$.

To show (iii) $\Rightarrow$ (iv), we construct a Kripke frame R with associated relational model Rel_R on the set of states

$$S \ \stackrel{\text{def}}{=} \ \mathsf{Traces}_G - (N_G(\,[\![usu]\!]_G) \cup C).$$

Note that for any $\sigma\tau\rho \in \mathsf{Traces}_G$, if $\tau \in N_G(\,[\![usu]\!]_G) \cup C$, then $\sigma\tau\rho \in N_G(\,[\![usu]\!]_G) \cup C$, so any subtrace of a trace in S is also in S. Moreover, any string with a $\rhd$-redex is in C, so every element of S is in $\rhd$-normal form.

Atomic symbols are interpreted in R as follows:

$$\begin{aligned}
\mathfrak{m}_R(p) &\stackrel{\text{def}}{=} \{(\sigma, \sigma N_G(\alpha p \beta)) \mid \sigma N_G(\alpha p \beta) \in S\}, \quad p \in \mathsf{P} \\
\mathfrak{m}_R(b) &\stackrel{\text{def}}{=} \{\sigma \in S \mid \textbf{last}(\sigma) \leq b\}, \quad b \in \mathsf{B}.
\end{aligned}$$

We now show that for all $t \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$,

$$[\,t\,]_R \ = \ \{(\sigma, \sigma\tau) \mid \sigma\tau \in S,\ \tau \in N_G(\,[\![t]\!]_G)\} \tag{7}$$

by induction on the structure of t. For $p \in \mathsf{P}$ and $b \in \mathsf{B}$,

$$\begin{aligned}
[\,p\,]_R &= \{(\sigma, \sigma N_G(\alpha p \beta)) \mid \sigma N_G(\alpha p \beta) \in S\} \\
&= \{(\sigma, \sigma\tau) \mid \sigma\tau \in S,\ \tau \in N_G(\mathfrak{m}_G(p))\} \\
&= \{(\sigma, \sigma\tau) \mid \sigma\tau \in S,\ \tau \in N_G(\,[\![p]\!]_G)\}, \\
[\,b\,]_R &= \{(\sigma, \sigma) \mid \sigma \in S,\ \textbf{last}(\sigma) \leq b\} \\
&= \{(\sigma, \sigma) \mid \sigma \in S,\ \textbf{last}(\sigma) \in \ [\![b]\!]_G\} \\
&= \{(\sigma, \sigma\tau) \mid \sigma\tau \in S,\ \tau \in N_G(\,[\![b]\!]_G)\}.
\end{aligned}$$

For the constants 0 and 1, we have

$$
\begin{aligned}
[0]_R &= \varnothing = \{(\sigma,\sigma\tau) \mid \sigma\tau \in S,\ \tau \in N_G([\![0]\!]_G)\}\\
[1]_R &= \{(\sigma,\sigma) \mid \sigma \in S\} = \{(\sigma,\sigma\tau) \mid \sigma\tau \in S,\ \tau \in N_G([\![1]\!]_G)\}.
\end{aligned}
$$

For compound expressions,

$$
\begin{aligned}
&[t_1+t_2]_R\\
&\quad= [t_1]_R \cup [t_2]_R\\
&\quad= \{(\sigma,\sigma\tau) \mid \sigma\tau \in S,\ \tau \in N_G([\![t_1]\!]_G)\} \cup \{(\sigma,\sigma\tau) \mid \sigma\tau \in S,\ \tau \in N_G([\![t_2]\!]_G)\}\\
&\quad= \{(\sigma,\sigma\tau) \mid \sigma\tau \in S,\ \tau \in N_G([\![t_1]\!]_G) \cup N_G([\![t_2]\!]_G)\}\\
&\quad= \{(\sigma,\sigma\tau) \mid \sigma\tau \in S,\ \tau \in N_G([\![t_1+t_2]\!]_G)\},
\end{aligned}
$$

$$
\begin{aligned}
[t_1t_2]_R &= [t_1]_R \circ [t_2]_R\\
&= \{(\sigma,\sigma\tau\rho) \mid (\sigma,\sigma\tau) \in [t_1]_R \wedge (\sigma\tau,\sigma\tau\rho) \in [t_2]_R\}\\
&= \{(\sigma,\sigma\tau\rho) \mid \sigma\tau\rho \in S,\ \tau \in N_G([\![t_1]\!]_G),\ \rho \in N_G([\![t_2]\!]_G)\}\\
&= \{(\sigma,\sigma\upsilon) \mid \sigma\upsilon \in S,\ \upsilon \in N_G([\![t_1]\!]_G)N_G([\![t_2]\!]_G)\} \quad \text{taking } \upsilon = \tau\rho\\
&= \{(\sigma,\sigma\upsilon) \mid \sigma\upsilon \in S,\ \upsilon \in N_G([\![t_1t_2]\!]_G)\},
\end{aligned}
$$

$$
\begin{aligned}
[t^*]_R &= \bigcup_n [t]_R^n\\
&= \bigcup_n \{(\sigma,\sigma\tau) \mid \sigma\tau \in S,\ \tau \in N_G([\![t^n]\!]_G)\}\\
&= \{(\sigma,\sigma\tau) \mid \sigma\tau \in S,\ \tau \in N_G([\![t^*]\!]_G)\},
\end{aligned}
$$

$$
\begin{aligned}
[\bar{b}]_R &= [1]_R - [b]_R\\
&= \{(\sigma,\sigma) \mid \sigma \in S\} - \{(\sigma,\sigma\tau) \mid \sigma\tau \in S,\ \tau \in N_G([\![b]\!]_G)\}\\
&= \{(\sigma,\sigma) \mid \sigma \in S\} - \{(\sigma,\sigma) \mid \sigma \in S,\ \textbf{last}(\sigma) \in [\![b]\!]_G\}\\
&= \{(\sigma,\sigma) \mid \sigma \in S,\ \textbf{last}(\sigma) \in [\![\bar{b}]\!]_G\}\\
&= \{(\sigma,\sigma\tau) \mid \sigma\tau \in S,\ \tau \in N_G([\![\bar{b}]\!]_G)\}.
\end{aligned}
$$

It follows from (7) that $[t_1]_R = [t_2]_R$ iff $N_G([\![t_1]\!]_G) \cap S = N_G([\![t_2]\!]_G) \cap S$. The direction $\Leftarrow$ is clear. Conversely, if $[t_1]_R = [t_2]_R$, then

$$
\begin{aligned}
N_G([\![t_1]\!]_G) \cap S &= \{\tau \mid (\textbf{first}(\tau),\tau) \in [t_1]_R\} \qquad \text{by (7)}\\
&= \{\tau \mid (\textbf{first}(\tau),\tau) \in [t_2]_R\}\\
&= N_G([\![t_2]\!]_G) \cap S.
\end{aligned}
$$

Now for $1 \leq i \leq n$, observe that $N_G([\![c_i r_i]\!]_G) \cap S = N_G([\![c_i]\!]_G) \cap S$ by considering the two types of strings in $[\![c_i r_i]\!]_G$, namely $\alpha r_i \alpha$ and $\alpha r_i \beta$ for atoms $\alpha \leq c_i$ and $\beta \neq \alpha$. The former reduce to $\alpha \in [\![c_i]\!]_G$ under $\rhd$, and the latter are in $\rhd$-normal form but not in S. It follows that $[c_i r_i]_R = [c_i]_R$.

Moreover, $N_G([\![s]\!]_G) \cap S \subseteq N_G([\![usu]\!]_G) \cap S = \varnothing = N_G([\![0]\!]_G) \cap S$, so $[s]_R = [0]_R$.

We have shown that

$$\mathsf{Rel}_R \quad \vDash \quad s = 0 \wedge \bigwedge_{i=1}^{n} c_i r_i = c_i,$$

therefore Rel_R satisfies all the premises of (iii) in the statement of the lemma. It follows from (iii) that $[p]_R = [q]_R$, from which we can conclude

$$N_G([\![p]\!]_G) \cap S = N_G([\![q]\!]_G) \cap S. \tag{8}$$

But

$$\begin{aligned}
[\![H(p + usu)]\!]_G &= h([\![p + usu]\!]_G) \quad \text{by Lemma 3.3} \\
&= (N_G([\![p]\!]_G) \cup N_G([\![usu]\!]_G)) - C \\
&= (N_G([\![p]\!]_G) - C - N_G([\![usu]\!]_G)) \cup (N_G([\![usu]\!]_G) - C) \\
&= (N_G([\![p]\!]_G) \cap S) \cup (N_G([\![usu]\!]_G) - C),
\end{aligned}$$

and similarly $[\![H(q + usu)]\!]_G = (N_G([\![q]\!]_G) \cap S) \cup (N_G([\![usu]\!]_G) - C)$, therefore by (8), $[\![H(p + usu)]\!]_G = [\![H(q + usu)]\!]_G$. Since Tr_G is the free KAT on generators P, B [16], we have $\vDash H(p + usu) = H(q + usu)$. This completes the proof of (iii) $\Rightarrow$ (iv).

Finally, to show (iv) $\Rightarrow$ (i), suppose $\vDash H(p + usu) = H(q + usu)$. Let I be an arbitrary interpretation over a Kleene algebra with tests K such that

$$K, I \quad \vDash \quad s = 0 \wedge \bigwedge_{i=0}^{n} c_i r_i = c_i.$$

By Lemma 2.1,

$$K, I \quad \vDash \quad \bigwedge_{i=0}^{n} r_i = \overline{c}_i r_i + c_i,$$

so $K, I \vDash H(t) = t$ for any $t \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$. Thus the following equations all hold under the interpretation I:

$$p \;=\; p + usu \;=\; H(p + usu) \;=\; H(q + usu) \;=\; q + usu \;=\; q.$$

Thus K, I satisfies the Horn formula of (i). Since K and I were arbirtrary, this formula holds in all Kleene algebras with tests. $\square$

We have proved Theorem 3.2 except for the complexity argument. The above transformation of our hypotheses can clearly be done in *PTIME*. In general, sequences of substitutions can cause exponential blowup in term size; for example,

$$a_1[a_1/a_2^2][a_2/a_3^2]\cdots[a_j/a_{j+1}^2] = a_{j+1}^{2^j}.$$

However, this cannot occur in our case because r_i does not appear in $\overline{c}_j r_j + c_j$ for $i \neq j$, and otherwise it is clear that the calculation of $\widehat{p} = H(p + usu)$ and $\widehat{q} = H(q + usu)$ is in *PTIME*. Note that this is relative to $s_1, \ldots, s_m$, $c_1, \ldots, c_n$, $r_1, \ldots, r_n$, p, q, and P. We must know P for the "$+\, usu$" part of $H(p + usu)$ and $H(q + usu)$.

In [6], it is shown that the equational theory of KAT is decidable in *PSPACE*. Because $\widehat{p}, \widehat{q}$ can be calculated in *PTIME*, (i)–(iii) are decidable in *PSPACE* as well.

Acknowledgments

This work was supported in part by NSF grant CCR-0105586 and ONR Grant N00014-01-1-0968. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of these organizations or the US Government.

References

[1] Allegra Angus and Dexter Kozen. Kleene algebra with tests and program schematology. Technical Report 2001-1844, Computer Science Department, Cornell University, July 2001.

[2] Adam Barth and Dexter Kozen. Equational verification of cache blocking in LU decomposition using Kleene algebra with tests. Technical Report 2002-1865, Computer Science Department, Cornell University, June 2002.

[3] Ernie Cohen. Hypotheses in Kleene algebra. Unpublished, April 1994.

[4] Ernie Cohen. Lazy caching. Unpublished, 1994.

[5] Ernie Cohen. Using Kleene algebra to reason about concurrency control. Unpublished, 1994.

[6] Ernie Cohen, Dexter Kozen, and Frederick Smith. The complexity of Kleene algebra with tests. Technical Report 96-1598, Computer Science Department, Cornell University, July 1996.

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

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

[9] Donald M. Kaplan. Regular expressions and the equivalence of programs. *J. Comput. Syst. Sci.*, 3:361–386, 1969.

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

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

[12] Dexter Kozen. Kleene algebra with tests. *Transactions on Programming Languages and Systems*, 19(3):427–443, May 1997.

[13] Dexter Kozen. On the complexity of reasoning in Kleene algebra. In *Proc. 12th Symp. Logic in Comput. Sci.*, pages 195–202, Los Alamitos, Ca., June 1997. IEEE.

[14] Dexter Kozen. On Hoare logic and Kleene algebra with tests. *Trans. Computational Logic*, 1(1):60–76, July 2000.

[15] Dexter Kozen and Maria-Cristina Patron. Certification of compiler optimizations using Kleene algebra with tests. In John Lloyd, Veronica Dahl, Ulrich Furbach, Manfred Kerber, Kung-Kiu Lau, Catuscia Palamidessi, Luis Moniz Pereira, Yehoshua Sagiv, and Peter J. Stuckey, editors, *Proc. 1st Int. Conf. Computational Logic (CL2000)*, volume 1861 of *Lecture Notes in Artificial Intelligence*, pages 568–582, London, July 2000. Springer-Verlag.

[16] Dexter Kozen and Frederick Smith. Kleene algebra with tests: Completeness and decidability. In D. van Dalen and M. Bezem, editors, *Proc. 10th Int. Workshop Computer Science Logic (CSL'96)*, volume 1258 of *Lecture Notes in Computer Science*, pages 244–259, Utrecht, The Netherlands, September 1996. Springer-Verlag.

[17] Dexter Kozen and Jerzy Tiuryn. Intuitionistic linear logic and partial correctness. In *Proc. 16th Symp. Logic in Comput. Sci. (LICS'01)*, pages 259–268. IEEE, June 2001.