Modularizing the Elimination of $r = 0$ in Kleene Algebra

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

April 20, 2005

Abstract

Given a universal Horn formula of Kleene algebra of the form $r = 0 \to s = t$, it is already known that we can efficiently construct an equation $s' = t'$ which is valid if and only if $r = 0 \to s = t$ is valid. This is an example of *elimination of hypotheses*, which is useful because the equational theory of Kleene algebra is decidable while the universal Horn theory is not. We show that hypotheses of the form $r = 0$ can still be eliminated in the presence of other hypotheses. This lets us extend any technique for eliminating hypotheses to include hypotheses of the form $r = 0$.

1 Introduction

Kleene algebra (KA) arises in many areas of computer science, such as automata theory, the design and analysis of algorithms, dynamic logic, and program semantics. Many of these applications are enhanced by using Kleene algebra with tests (KAT), which combines KA with Boolean algebra.

We can use KAT to reason propositionally about programs (see [1, 13] for examples). The equivalence of an optimized and unoptimized program, the equivalence of an annotated and unannotated program, and partial correctness assertions can all be expressed as equations. The equational theory of KAT is well understood and has many useful properties; in particular, it is decidable (in *PSPACE*) and the theory remains unchanged when we restrict to relational interpretations [4, 14]. (Relational interpretations are of the greatest interest because the intended semantics are generally relational.)

However, we frequently wish to reason about programs under certain assumptions about the interaction of atomic programs and tests. For example, if p is the program "x := 0" and b is the assertion "x = 0", then we want to be able to make use of the facts $pb = p$ ("after running p, test b always succeeds") and $bp = b$ ("after test b succeeds, p is redundant") when reasoning about programs in which p and b appear; for instance, the equation $p^2 = p$ is not valid in KAT, but the formula $(pb = p \ \wedge \ bp = b) \rightarrow p^2 = p$ is. Thus, the *universal Horn theory* is of interest. A *univeral Horn formula* is an implication $E \rightarrow s = t$, where E is a finite set of equations. The word "universal" refers to the fact that the atomic symbols of E, s, and t are implicitly universally quantified. The *universal Horn theory* of a class of structures C, denoted $\mathcal{H}\mathsf{C}$, is the set of universal Horn formulas valid under all interpretations over structures in C.

The increased generality of the universal Horn theory is accompanied by greater complexity, and the theory does not remain the same when we restrict to important classes of Kleene algebras such as $*$-continuous Kleene algebras with tests (KAT^*) and relational Kleene algebras with tests (RKAT). $\mathcal{H}\mathsf{KAT}$ is Σ^0_1-complete (undecidable), $\mathcal{H}\mathsf{KAT}^*$ and $\mathcal{H}\mathsf{RKAT}$ are Π^1_1-complete (highly undecidable), and we have proper inclusions $\mathcal{H}\mathsf{KAT} \subsetneq \mathcal{H}\mathsf{KAT}^* \subsetneq \mathcal{H}\mathsf{RKAT}$ [12, 8].

Although these Horn theories are very complex in general, there are fragments of them that are both practical and of lower complexity. Consider the following theorem, fundamentally due to Cohen [2] and extended to the form below by Kozen and Smith [14, 11]. (The statement uses some notions that will not be defined until later, but we only need a vague understanding of it here.)

Theorem 1 *Let $r, s, t \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$, and let $u \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$ be the universal regular expression. Then the following are equivalent.*

$$\mathsf{KAT} \models r = 0 \rightarrow s = t \tag{1}$$
$$\mathsf{KAT}^* \models r = 0 \rightarrow s = t \tag{2}$$
$$\mathsf{RKAT} \models r = 0 \rightarrow s = t \tag{3}$$
$$\mathsf{KAT} \models s + uru = t + uru \tag{4}$$

The primary consequence of this theorem is that the Horn theory of Kleene algebra, restricted to formulas with hypotheses of the form $r = 0$, is decidable, and remains unchanged if we restrict to $*$-continuous or relational algebras: to decide if $r = 0 \rightarrow s = t$ is valid, we simply decide if $s + uru = t + uru$ is valid. In this way, we say that we have *eliminated* the hypothesis $r = 0$. It is also possible to eliminate other forms of hypotheses [2, 7].

The case $r = 0$ has particular significance, because partial correctness assertions can be expressed in KAT with equations of the form $r = 0$ (and multiple equations $r_1 = 0 \wedge \cdots \wedge r_k = 0$ can be combined into $r_1 + \cdots + r_k = 0$). So Theorem 1 shows that the Horn theory of KAT, restricted to hypotheses of the form $r = 0$, subsumes propositional Hoare logic, is decidable, and is furthermore complete for relational interpretations [11].

Our main result, Theorem 16 (p. 12), improves Theorem 1 so that $r = 0$ can be eliminated in the presence of other hypotheses. This allows any other technique for eliminating hypotheses to be extended to include $r = 0$. For example, if we have a technique for eliminating $f = g$ alone, we can eliminate $f = g \wedge r = 0$ by first eliminating $r = 0$ using Theorem 16, leaving hypothesis $f = g$, which can then be eliminated. In this way, Theorem 16 is like a module for eliminating $r = 0$ that can be added on to any other technique for eliminating hypotheses.

A related result, Corollary 24, shows that hypotheses of the form $cp = c$ (where c is Boolean and p is atomic) can be eliminated in the presence of other hypotheses, although the remaining hypotheses are modified. Hypotheses of the form $cp = c$ are useful for eliminating redundant code (consider our example $bp = b$ above; it expresses the fact that p is redundant when b already holds). (The procedure for eliminating $cp = c$ was introduced in [7], where it was shown how to eliminate $cp = c$ and $r = 0$ at the same time. Without the benefit of Theorem 16, this required a construction that simultaneously dealt with both $cp = c$ and $r = 0$.)

2 Preliminaries

For a more complete introduction to Kleene algebra and Kleene algebra with tests, see [10].

2.1 Kleene Algebra

Definition 2 *An* idempotent semiring *is a structure* $(S, +, \cdot, 0, 1)$ *satisfying*

$$\begin{aligned} x + x &= x \quad \textit{(idempotence)} \\ x + 0 &= x \\ x + y &= y + x \\ x + (y + z) &= (x + y) + z \\ 0 \cdot x &= x \cdot 0 = 0 \\ 1 \cdot x &= x \cdot 1 = x \\ x \cdot (y \cdot z) &= (x \cdot y) \cdot z \\ x \cdot (y + z) &= x \cdot y + x \cdot z \\ (y + z) \cdot x &= y \cdot x + z \cdot x \ . \end{aligned}$$

(In other words, $(S, +, 0)$ *is an upper semilattice with bottom element* 0*,* $(S, \cdot, 1)$ *is a monoid,* 0 *is an annihilator for* $\cdot$*, and* $\cdot$ *distributes over* $+$ *on the right and left.)*

We often write xy for $x \cdot y$. The upper semilattice structure induces a natural partial order on any idempotent semiring: $x \leq y \Leftrightarrow x + y = y$.

Definition 3 *A* Kleene algebra *is a structure* $(K, +, \cdot, ^*, 0, 1)$ *such that* $(K, +, \cdot, 0, 1)$

forms an idempotent semiring, and which satisfies

$$1 + xx^* \leq x^* \tag{5}$$

$$1 + x^*x \leq x^* \tag{6}$$

$$p + qx \leq x \rightarrow q^*p \leq x \tag{7}$$

$$p + xq \leq x \rightarrow pq^* \leq x \ . \tag{8}$$

(The order of precedence among the operators is $^* > \cdot > +$*, so that* $p + qr^* = p + (q \cdot (r^*))$*.) We let* KA *denote the category of all Kleene algebras and their homomorphisms. Equations (5)–(8) are called the* Kleene algebra $*$-axioms*.*

Given a set Σ *of constant symbols, let* RExp_Σ *be the set of Kleene algebra terms over* Σ*. We call the elements of* RExp_Σ regular expressions*, and the elements of* Σ atomic program symbols*. An* interpretation *is a homomorphism* $I : \mathsf{RExp}_\Sigma \rightarrow K$*, where* K *is a Kleene algebra.* I *is determined uniquely by its values on* Σ*.*

Equations (5) and (7) say that q^*p is the least solution of $p + qx \leq x$, while (6) and (8) say that pq^* is the least solution to $p + xq \leq x$.

A straightforward and vital consequence of the KA axioms[1] is that the operations $+$, $\cdot$, and * are monotone: if $x_0 \leq x_1$ and $y_0 \leq y_1$, then $x_0 + y_0 \leq x_1 + y_1$, $x_0y_0 \leq x_1y_1$, and $x_0^* \leq x_1^*$.

We use $\models$ to denote ordinary Tarskian satisfaction. However, since we have constant symbols from Σ not in the signatures of the underlying algebras, we will pair each algebra with an interpretation when speaking about satisfaction. For example, given a Kleene algebra K, interpretation $I : \mathsf{RExp}_\Sigma \rightarrow K$, and formula φ whose atomic program symbols are among Σ, we will write $K, I \models \varphi$ to indicate that K satisfies φ when the symbols in Σ are evaluated according to I. $K \models \varphi$ means that $K, I \models \varphi$ for every interpretation $I : \mathsf{RExp}_\Sigma \rightarrow K$. We also use $\models$ in two other standard ways: for a class C of algebras, $\mathsf{C} \models \varphi$ means that $K \models \varphi$ for each $K \in \mathsf{C}$; for a set Φ of formulas, $\Phi \models \varphi$ means that $K \models \varphi$ for each algebra K satisfying every formula in Φ.

We now introduce two particularly important types of Kleene algebras: *language algebras* and *relational algebras*.

Definition 4 *For an arbitrary monoid* M*, its powerset* 2^M *forms a Kleene algebra as follows.*

$$\begin{array}{rcl} 0 & = & \varnothing \\ 1 & = & \{1^M\} \quad \textit{(where } 1^M \textit{ is the identity of } M\textit{)} \\ A + B & = & A \cup B \\ A \cdot B & = & \{xy \ \mid \ x \in A, \ y \in B\} \\ A^* & = & \bigcup_{k \in \omega} A^k \end{array}$$

[1] The names of the categories we consider serve as convenient abbreviations for the type of algebra they contain. So, for example, "the KA axioms" means "the axioms of Kleene algebra".

We let $\mathsf{REG}\ M$ *denote the smallest subalgebra of* 2^M *containing the singletons* $\{x\}$, $x \in M$. *(The elements of* $\mathsf{REG}\ M$ *are the* regular subsets *of* M.*)* 2^M *and its subalgebras are known as* language algebras.

Of particular interest is the case $M = \Sigma^*$, *the monoid of all strings over alphabet* Σ *under concatenation. The empty string* ε *is the identity of this monoid. We define the canonical interpretation* $R : \mathsf{RExp}_\Sigma \to \mathsf{REG}\ \Sigma^*$ *by letting* $R(p) = \{p\}$ *(and extending* R *homomorphically to the rest of* RExp_Σ*). Note that we can interpret elements of* Σ^* *as elements of* RExp_Σ *in the obvious fashion.*

Definition 5 *For an arbitrary set* X, *the set* $2^{X\times X}$ *of all binary relations on* X *forms a Kleene algebra* $\mathcal{R}(X)$ *as follows.*

$$\begin{array}{rcl} 0 & = & \varnothing \\ 1 & = & \iota_X = \{(x,x) \mid x \in X\} \\ S+T & = & S \cup T \\ S \cdot T & = & S \circ T \quad \textit{(the composition of } S \textit{ with } T\textit{)} \\ S^* & = & \bigcup_{k\in\omega} S^k \quad \textit{(the reflexive transitive closure of } S\textit{)} \end{array}$$

A Kleene algebra K *is* relational *if it is a subalgebra of* $\mathcal{R}(X)$ *for some* X*;* X *is called the* base *of* K. *We let* RKA *denote the category of all relational Kleene algebras and their homomorphisms.*

The definitions of * in 2^M and $\mathcal{R}(X)$ exemplify the most common intuition about the meaning of *, which is that $y^* = \sup_{n\in\omega} y^n$, or informally, $y^* = 1 + y + y^2 + \cdots$. (More generally, if we require that multiplication distributes over this supremum, we have $xy^*z = x1z + xyz + xy^2z + \cdots = \sup_{n\in\omega} xy^nz$.) However, this property of * does not follow from the KA $*$-axioms, and must be postulated separately.

Definition 6 *A Kleene algebra* K *is* $*$-continuous *if it satisfies*

$$xy^*z = \sup_{k\in\omega} xy^kz$$

for all $x, y, z \in K$. *We let* KA^* *denote the category of all* $*$*-continuous Kleene algebras and their homomorphisms.*

Since relational composition distributes over arbitrary union, it is immediate from the definition of * in $\mathcal{R}(X)$ that relational Kleene algebras are $*$-continuous, so $\mathsf{RKA} \subseteq \mathsf{KA}^*$.

The following ubiquitous lemma is a useful generalization of $*$-continuity.

Lemma 7 *Suppose* $K \in \mathsf{KA}^*$, $I : \mathsf{RExp}_\Sigma \to K$ *is an interpretation, and* $t \in \mathsf{RExp}_\Sigma$. *Then*

$$I(t) = \sup_{\sigma\in R(t)} I(\sigma) \ .$$

Proof By induction on structure of t. For details, see [9, Lemma 7.1, pp. 246–248]. $\square$

2.2 Kleene Algebra with Tests

We can combine Kleene algebra with Boolean algebra to get *Kleene algebra with tests*. The Boolean aspect is useful for capturing Boolean aspects of programming semantics, particularly control flow and assertions.

Definition 8 *A* Kleene algebra with tests *is a two-sorted structure* $(K, B, +, \cdot, ^*, \overline{}, 0, 1)$*, where* $(K, +, \cdot, ^*, 0, 1)$ *is a Kleene algebra, and* $(B, +, \cdot, \overline{}, 0, 1)$ *is a Boolean subalgebra. The elements of* B *are called* tests. *We let* KAT *denote the category of all Kleene algebras with tests and their homomorphisms; we let* KAT^* *denote the subcategory of all* $*$*-continuous Kleene algebras with tests.*

We now have two types of atomic symbols: programs and tests. For a finite set P *of atomic program symbols and a finite set* B *of atomic test symbols,* $\mathsf{RExp}_{\mathsf{P},\mathsf{B}}$ *is the set of* KAT *terms over* P *and* B*; negation can only be applied to Boolean terms, which are terms built from* $0,1,+,\cdot,\overline{}$*, and atomic test symbols. An interpretation* $I : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \to K$ *must map each atomic test to a test in* K *(and it follows by induction that it will map all Boolean terms to tests).*

$\mathcal{R}(X)$ *forms a Kleene algebra with tests by keeping the previously defined Kleene algebra structure, and letting* $B = \{r \in \mathcal{R}(X) \mid r \leq 1\}$, $\overline{b} = \iota_X - b$*. A Kleene algebra with tests* K *is relational if it is a subalgebra of* $\mathcal{R}(X)$ *for some* X*. We let* RKAT *denote the category of all relational Kleene algebras with tests and their homomorphisms.*

Every Kleene algebra induces a Kleene algebra with tests by letting $B = \{0, 1\}$, the two-element Boolean algebra; conversely, every Kleene algebra with tests induces a Kleene algebra by taking its reduct to the signature of Kleene algebra (*i.e.*, taking its image under the map $(K, B, +, \cdot, ^*, \overline{}, 0, 1) \mapsto (K, +, \cdot, ^*, 0, 1)$). With this in mind, it is easy to see that for any formula φ in the language of Kleene algebra, $\mathsf{KAT} \models \varphi \Leftrightarrow \mathsf{KA} \models \varphi$, $\mathsf{KAT}^* \models \varphi \Leftrightarrow \mathsf{KA}^* \models \varphi$, and $\mathsf{RKAT} \models \varphi \Leftrightarrow \mathsf{RKA} \models \varphi$.

There is an analog of $\mathsf{REG}\ \Sigma^*$ for KAT called the *guarded-string model*, with its own analog of the canonical interpretation R. Though the guarded-string model is in general very important for studying KAT, we will not need it for our results here, and refer the reader to [14] for further information on guarded strings.

The following elementary lemma about subalgebras will be needed in Lemma 17.

Lemma 9 *Let* $K \in \mathsf{KA}$ *and let* $x \in K$*. Then* $\{y \in K \mid y \leq x\}$ *is a subalgebra of* K *iff* $x = y^*$ *for some* $y \in K$ *(or equivalently,* $x = x^*$*). The same also holds for* KAT*s. (Note that this is not claiming that all subalgebras of* K *have this form.)*

The proof is straightforward and may safely be skipped.

Proof Let $K' = \{y \in K \mid y \leq x\}$.

Suppose K' is a subalgebra of K. Then $x^* \in K'$, so $x^* \leq x$, so $x = x^*$.

Suppose $x = y^*$ for some $y \in K$. Then $x^* = y^{**} = y^* = x$. The necessary closure conditions follow from monotonicity and the fact that $0+1+xx+(x+x)+x^* \leq x^*$. (For example, for any $y_1, y_2 \in K'$, we have $y_1 y_2 \leq xx \leq x^*$.) $\square$

2.3 Universal Horn Formulas

Definition 10 *A* universal Horn formula *is a formula of the form*

$$s_1 = t_1 \wedge \cdots \wedge s_t = t_k \rightarrow s = t \ ,$$

where s_i, t_i, s, t are terms. The set of universal Horn formulas valid over a class C *of algebras is the* universal Horn theory *of* C*, which we denote by* $\mathcal{H}\mathsf{C}$*.*

We will often drop the word "universal". We will allow finite sets of equations to appear in the hypotheses of a Horn formula, by taking their conjunction; *e.g.*, if $E = \{pq = qp,\ p \leq 1\}$, then $E \rightarrow s = t$ means $(pq = qp \wedge p \leq 1) \rightarrow s = t$.

Lemma 11 *Let Γ be any class of $*$-continuous Kleene algebras with interpretations. (That is, Γ consists of pairs (K, I) where $K \in \mathsf{KA}^*$ and $I : \mathsf{RExp}_\Sigma \rightarrow K$ is an interpretation.) Then for any Horn formula of the form $E \rightarrow s \leq t$,*

$$\Gamma \models E \rightarrow s \leq t \iff (\forall \sigma \in R(s)) \ \ \Gamma \models E \rightarrow \sigma \leq t \ .$$

Proof For any $K \in \mathsf{KA}^*$ with interpretation $I : \mathsf{RExp}_\Sigma \rightarrow K$, the equivalence

$$K, I \models E \rightarrow s \leq t \iff (\forall \sigma \in R(s)) \ \ K, I \models E \rightarrow \sigma \leq t$$

is a straightforward consequence of Lemma 7. The lemma then follows by exchanging the universal quantifiers $(\forall \sigma \in R(s))$ and $(\forall (K, I) \in \Gamma)$. (This latter quantifier comes from $\Gamma \models E \rightarrow s \leq t \Leftrightarrow (\forall (K, I) \in \Gamma)\ K, I \models E \rightarrow s \leq t$.) $\square$

2.4 A Proof System for $\mathcal{H}\mathsf{RKA}$

Later, in the proof of Lemma 19, we will use a proof-theoretic argument based on the infinitary proof system for $\mathcal{H}\mathsf{RKA}$ introduced in [6]. We will only present the material that we will need in Section 3.1 for the proof of Lemma 19; for a more thorough treatment, please see [6].

2.4.1 Finite Automata and Trees

Our proof system for $\mathcal{H}\mathsf{RKA}$ is based on trees of finite automata, and we must define a number of notions related to trees and automata before continuing.

Assume we have a fixed finite alphabet Σ. We let NFA denote the set of all nondeterministic finite automata over Σ, requiring that all states be natural numbers; ε-moves (also called ε-edges) are allowed.

We will also use NFA as shorthand for *nondeterministic finite automaton.* For any NFA A, $L(A)$ denotes the language of A, and $|A|$ denotes the states of A. For states $v, w \in |A|$, let $A^{v,w}$ denote the NFA which is identical to A except that it has v and w as its unique start and accept states, respectively. We fix distinct natural numbers a and b (say, $a = 0$ and $b = 1$), and let $\mathrm{NFA}^{a,b}$ be the set of all $A \in \mathrm{NFA}$ which have unique start state a and unique accept state b.

We define $F_0 \in \mathrm{NFA}^{a,b}$ to have states $\{a, b\}$ and no edges.

Given an NFA A and states $v, w \in |A|$, we will sometimes want to "insert" a string $\tau \in \Sigma^*$ into $L(A^{v,w})$. For this purpose, we define $A' = \mathrm{insert}_2(A, v, w, \tau)$ as follows.

1. If $\tau = p_1 \cdots p_k$, with $p_i \in \Sigma$ and $k > 0$, we obtain A' from A by adding $k - 1$ new states $x_1, \ldots, x_{k-1}$ and adding edges
$$v \xrightarrow{p_1} x_1 \xrightarrow{p_2} \cdots \xrightarrow{p_{k-1}} x_{k-1} \xrightarrow{p_k} w \ .$$
2. If $\tau = \varepsilon$, then we add an ε-edge from v to w and also from w to v. (Where it is used, $\mathrm{insert}_2(A, v, w, \varepsilon)$ corresponds to identifying v and w with each other. The edge from w to v, called a *reverse ε-edge*, is needed to capture the symmetry of the identity relation.)

We now move on to trees. ω is the set of naturals $\{0, 1, \ldots\}$. $\omega^{<\omega}$ is the set of all finite strings of naturals (including the empty string). A set $T \subseteq \omega^{<\omega}$ is a *tree* if it is closed under taking initial segments. A function $f : \omega \to \omega$ can be treated as an infinite sequence of naturals, and for $n \in \omega$, we let $f \restriction n$ denote the initial segment of f of length n. Such an f is a *path* through a tree T if $(f \restriction n) \in T$ for all $n \in \omega$.

2.4.2 Relational Proofs

The following definition of *relational proof* captures, with trees of finite automata, the combinatorics of attempting to construct a relational counterexample to a Horn formula. A path through such a tree yields a relational model in which the formula fails, while well-foundedness establishes the impossibility of a counterexample (*i.e.*, the relational validity of the formula).

Definition 12 *Let $E \to \sigma \leq t$ be a Horn formula in the language of KA with $\sigma \in \Sigma^*$ and $t \in \mathsf{RExp}_\Sigma$. We assume that all hypotheses in E are inequalities $x \leq y$, by breaking any equations $x = y$ into $x \leq y \wedge y \leq x$ as necessary. We fix distinct states a and b as above. We fix a special symbol* CON, *which will signify contradiction.*

A relational tree *for $E \to \sigma \leq t$ is a pair (T, A) where $T \subseteq \omega^{<\omega}$ is a tree and $A : T \to \mathrm{NFA}^{a,b} \cup \{\mathrm{CON}\}$ such that the following conditions hold. (A_f will denote $A(f)$.)*

1. *At the root, we have* $A_{\langle\rangle} = \mathrm{insert}_2(F_0, a, b, \sigma)$.

2. $f \in T$ *is a leaf node if and only if* $A_f = \mathrm{CON}$ *or* $R(t) \cap L(A_f) \neq \varnothing$.

3. *If* f *is not a leaf node, then there exist* $v, w \in |A_f|$ *(possibly equal),* $\rho \in L(A_f^{v,w})$*, and an inequality* $r \leq r'$ *in* E *such that* $\rho \in R(r)$ *and*

   (a) *if* $R(r') = \varnothing$ *(typically because* $r' = 0$*), then* f *has one child* g*, with* $A_g = \mathrm{CON}$*;*

   (b) *if* $R(r') \neq \varnothing$*, then* f *has one child* g_τ *for each* $\tau \in R(r')$*, with* $A_{g_\tau} = \mathrm{insert}_2(A_f, v, w, \tau)$.

   (We say that the hypothesis $r \leq r'$ *is* applied at f*.)*

A relational proof of $E \to \sigma \leq t$ *is a well-founded relational tree for* $E \to \sigma \leq t$*. We say* $E \to \sigma \leq t$ *is* relationally provable *if such a proof exists.*

Lemma 13 *For any Horn formula of the form* $E \to \sigma \leq t$*, the following are equivalent.*

(i) $\mathsf{RKA} \models E \to \sigma \leq t$

(ii) $E \to \sigma \leq t$ *is relationally provable.*

Proof See [5] or [6]. □

The notion of relational provability can be extended to arbitrary Horn formulas, but we will not need it for the proof of Lemma 19.

2.5 The Relationship Between $\mathcal{H}$RKA and $\mathcal{H}$RKAT

The system presented in Section 2.4 is a tool for studying $\mathcal{H}$RKA, while in Lemma 19, we will wish to use it to draw conclusions about $\mathcal{H}$RKAT. This must be rectified, and there are multiple ways to proceed. One would be to modify the notion of relational proof so that it applies to $\mathcal{H}$RKAT; this would present no particular difficulty, but would require a closer look at relational proofs than we would like to get into here. Instead, we will show how to reduce questions about $\mathcal{H}$RKAT to $\mathcal{H}$RKA, in a way that will allow us to use the existing definition of relational proof when proving Lemma 19.

Lemma 14 *For any Horn formula* φ *of* KAT*, there is a Horn formula* $\mathsf{Tr}(\varphi)$ *of* KA *such that* $\mathsf{RKAT} \models \varphi$ *iff* $\mathsf{RKA} \models \mathsf{Tr}(\varphi)$.

The truth of the lemma is uninteresting without putting restrictions on the nature of the translation Tr. However, instead of trying to capture the desired properties of Tr for inclusion in the lemma, we just give the proof, and observe later that the translation works for a particular purpose when the need arises.

Proof (Outline: we first assume that negation is only applied to atomic tests, then replace the negations of atomic tests with fresh program symbols, and finally add new hypotheses to ensure that the new program symbols behave like the negated tests they replace.)

Fix a set P of atomic program symbols, and a set B of atomic tests. Given any $s \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$, we can assume without loss of generality that negation is only applied to atomic tests, in light of DeMorgan's Laws. To make this precise, we define $\nu : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \to \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$ by recursion:

$$
\begin{array}{rcl}
\nu(t_1 + t_2) & = & \nu(t_1) + \nu(t_2) \\
\nu(t_1 \cdot t_2) & = & \nu(t_1) \cdot \nu(t_2) \\
\nu(t^*) & = & \nu(t)^* \\
\nu(\overline{\overline{t}}) & = & t \\
\nu(\overline{t_1 \cdot t_2}) & = & \nu(\overline{t_1}) + \nu(\overline{t_2}) \\
\nu(\overline{t_1 + t_2}) & = & \nu(\overline{t_1}) \cdot \nu(\overline{t_2}) \\
\nu(\overline{0}) & = & 1 \\
\nu(\overline{1}) & = & 0 \\
\nu(\overline{b}) & = & \overline{b} \quad (b \in \mathsf{B}) \\
\nu(p) & = & p \quad (p \in \mathsf{P} \bigcup \mathsf{B} \bigcup \{0, 1\})
\end{array}
$$

For any $t \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$, $\mathsf{KAT} \models t = \nu(t)$, and in $\nu(t)$, negation is only applied to atomic tests.

For each $b \in \mathsf{B}$, we introduce two new atomic program symbols $\tilde{b}$ and $\tilde{\overline{b}}$, and we let $\Sigma = \mathsf{P} \bigcup \{\tilde{b}, \tilde{\overline{b}} \;\mid\; b \in \mathsf{B}\}$. For any $t \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$, we let $\tilde{t}$ be the result of taking $\nu(t)$, and replacing all occurrences of $\overline{b}$ with $\tilde{\overline{b}}$, and all positive occurrences of b with $\tilde{b}$ (for each $b \in \mathsf{B}$). Note that $\tilde{t} \in \mathsf{RExp}_{\Sigma}$. For any formula φ, we let $\tilde{\varphi}$ be the result of replacing each term t in φ with $\tilde{t}$.

Now take any Horn formula φ of the form $\theta \to \psi$ (with all terms in $\mathsf{RExp}_{\mathsf{P},\mathsf{B}}$). Let $\mathsf{Tr}(\varphi)$ be the formula

$$\left(\tilde{\theta} \wedge \bigwedge_{b \in \mathsf{B}} (\tilde{b} + \tilde{\overline{b}} = 1 \wedge \tilde{b} \cdot \tilde{\overline{b}} = 0)\right) \to \tilde{\psi} \;.$$

(The extra hypotheses make $\tilde{b}$ and $\tilde{\overline{b}}$ behave like Boolean complements of each other.)

We now show $\mathsf{RKAT} \models \varphi$ iff $\mathsf{RKA} \models \mathsf{Tr}(\varphi)$.

Suppose $\mathsf{RKAT} \not\models \varphi$. Let $K \in \mathsf{RKAT}$ with interpretation $I : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \to K$ such that $K, I \not\models \varphi$. Then $K, I \models \theta \wedge \neg\psi$. Define the interpretation $\tilde{I} : \mathsf{RExp}_{\Sigma} \to K$ by

$$\tilde{I}(p) = \begin{cases} I(p), & \text{if } p \in \mathsf{P}, \\ I(b), & \text{if } p = \tilde{b}, \\ I(\overline{b}), & \text{if } p = \tilde{\overline{b}}. \end{cases}$$

A simple induction shows that for any $t \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$, $\tilde{I}(\tilde{t}) = I(\nu(t)) = I(t)$. It follows that $K, \tilde{I} \models \tilde{\theta} \wedge \neg\tilde{\psi}$, since $K, I \models \theta \wedge \neg\psi$. Also,

$$K, \tilde{I} \models \bigwedge_{b \in \mathsf{B}} (\tilde{b} + \tilde{\bar{b}} = 1 \wedge \tilde{b} \cdot \tilde{\bar{b}} = 0) \ .$$

Thus $K, \tilde{I} \not\models \mathsf{Tr}(\varphi)$, so $\mathsf{RKA} \not\models \mathsf{Tr}(\varphi)$ (recall that we can treat K as a member of RKA by passing it through the forgetful functor which drops negation). Therefore, $\mathsf{RKA} \models \mathsf{Tr}(\varphi) \rightarrow \mathsf{RKAT} \models \varphi$.

Now suppose that $\mathsf{RKA} \not\models \mathsf{Tr}(\varphi)$. Let $K \in \mathsf{RKA}$ with interpretation $I : \mathsf{RExp}_\Sigma \rightarrow K$ such that $K, I \not\models \mathsf{Tr}(\varphi)$. Let X be the base of K. Then $K \subseteq \mathcal{R}(X)$, so $\mathcal{R}(X), I \not\models \mathsf{Tr}(\varphi)$; that is,

$$\mathcal{R}(X), I \models \tilde{\theta} \wedge \bigwedge_{b \in \mathsf{B}} (\tilde{b} + \tilde{\bar{b}} = 1 \wedge \tilde{b} \cdot \tilde{\bar{b}} = 0) \wedge \neg\tilde{\psi} \ .$$

In particular, for any $b \in \mathsf{B}$, $I(\tilde{b}) \bigcup I(\tilde{\bar{b}}) = I(1)$, and $I(\tilde{b}) \circ I(\tilde{\bar{b}}) = \varnothing$; it follows that $I(\tilde{b}) \bigcap I(\tilde{\bar{b}}) = \varnothing$ (since $R \bigcap S = R \circ S$ whenever $R, S \subseteq I(1)$), so $I(\tilde{\bar{b}}) = I(1) - I(\tilde{b})$.

Define the interpretation $I' : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \rightarrow \mathcal{R}(X)$ by

$$\begin{aligned} I'(p) &= I(p) \ , \\ I'(b) &= I(\tilde{b}) \ . \end{aligned}$$

We have

$$\begin{aligned} I'(\bar{b}) &= I'(1) - I'(b) \\ &= I(1) - I(\tilde{b}) \\ &= I(\tilde{\bar{b}}) \ . \end{aligned}$$

It follows that, for any $t \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$, $I'(t) = I(\tilde{t})$. So, $\mathcal{R}(X), I' \models \theta \wedge \neg\psi$, since $\mathcal{R}(X), I \models \tilde{\theta} \wedge \neg\tilde{\psi}$, giving us $\mathsf{RKAT} \not\models \varphi$. Therefore, $\mathsf{RKAT} \models \varphi \rightarrow \mathsf{RKA} \models \mathsf{Tr}(\varphi)$, completing the proof. □

3 Main results

3.1 Eliminating $r = 0$

Definition 15 *For a fixed set $\mathsf{P} = \{p_1, \ldots, p_n\}$ of atomic program symbols, the* universal regular expression *u is defined by*

$$u = (p_1 + \cdots + p_n)^* \ .$$

We trivially have $\mathsf{KAT} \models u = uu = u^*$, and a straightforward induction shows that, for any $s \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$, $\mathsf{KAT} \models s \leq u$.

Our goal is the following theorem.

Theorem 16 *Let u be the universal regular expression, let E be any finite set of hypotheses, and let $r, s, t \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$. Then the following equivalences hold.*

$$\mathsf{KAT} \models E \wedge r = 0 \to s = t \iff \mathsf{KAT} \models E \to s + uru = t + uru \quad (9)$$
$$\mathsf{KAT}^* \models E \wedge r = 0 \to s = t \iff \mathsf{KAT}^* \models E \to s + uru = t + uru \quad (10)$$
$$\mathsf{RKAT} \models E \wedge r = 0 \to s = t \iff \mathsf{RKAT} \models E \to s + uru = t + uru \quad (11)$$

Note that the special case $E = \varnothing$ is essentially Theorem 1 (when $E = \varnothing$, the right hand sides of (9)–(11) are equivalent, since the equational theories of KAT, KAT^*, and RKAT coincide). Note also that for any formula φ in the language of KA, we have $\mathsf{KA} \models \varphi$ iff $\mathsf{KAT} \models \varphi$, $\mathsf{KA}^* \models \varphi$ iff $\mathsf{KAT}^* \models \varphi$, *etc.*, so Theorem 16 also applies to KA, KA^*, and RKA. (Alternatively, omitting the Boolean aspects of the proof that follows would yield a proof of the analogous theorem for KA, KA^*, and RKA.)

We prove each equivalence as a separate lemma. Fix u, E, r, s, t, as above.

Lemma 17

$$\mathsf{KAT} \models E \wedge r = 0 \to s = t \iff \mathsf{KAT} \models E \to s + uru = t + uru$$

Proof The right-to-left implication is trivial: reasoning under $E \wedge r = 0$, we have $s = s + 0 = s + uru = t + uru = t + 0 = t$. (Note that this argument also applies to KAT^* and RKAT.)

For the left-to-right implication, suppose $\mathsf{KAT} \models E \wedge r = 0 \to s = t$. Take any $K \in \mathsf{KAT}$ with interpretation I such that $K, I \models E$. Let $\perp = I(uru)$, $\top = I(u)$, noting that $\top^* = \top$, $\perp = \top\perp = \perp\top$, and $\perp\perp \leq \perp$. Let $K' = \{x \in K \mid x \leq \top\}$. This is a subalgebra of K by Lemma 9, since $\top = \top^*$. I is an interpretation into K'.

Define the map $f : K' \to K'$ by $f(x) = x + \perp$. Let $L = f[K']$, the image of K' under f. $\top$ and $\perp$ are respectively the greatest and least elements of L. Note that for any $x \in K'$, $x\top \leq \top$, so $x\perp = x\top\perp \leq \top\perp = \perp$. We similarly have $\perp x \leq \perp$.

Define

$$\begin{aligned} 0^L &= \perp = f(0) \\ 1^L &= 1 + \perp = f(1) \\ v \cdot^L w &= v \cdot w + \perp \ . \end{aligned}$$

Let L be the structure $(L, f[B], +, \cdot^L, {}^*, \tilde{}, 0^L, 1^L)$, in the signature of KAT, where B is the set of tests of K', and the Boolean complement $\tilde{}$ is defined by $\widetilde{f(c)} = f(\overline{c})$.

We must show that $\tilde{}$ is well-defined. Suppose $f(c) = f(d)$. Then

$$\begin{array}{rcll} f(\overline{c}) & = & \overline{c} + \bot & \\ & \leq & (\overline{c} + \bot)(1 + \bot) & \\ & = & (\overline{c} + \bot)(d + \overline{d} + \bot) & \\ & = & (\overline{c} + \bot)(c + \overline{d} + \bot) & (\text{since } c + \bot = f(c) = f(d) = d + \bot) \\ & = & \overline{c}c + \overline{c}\overline{d} + \overline{c}\bot + \bot c + \bot\overline{d} + \bot\bot & \\ & \leq & 0 + \overline{d} + \bot & \\ & = & f(\overline{d}) \ . & \end{array}$$

Similarly, $f(\overline{d}) \leq f(\overline{c})$, so $f(\overline{c}) = f(\overline{d})$. Therefore, $\tilde{}$ is well-defined.

We claim that $f : K' \to L$ is a homomorpishm. (Note that this is different from claiming that $f : K' \to K'$ is a homomorphism, which is not true unless $\bot = 0$.) For any $x, y \in K$, and c a test in K,

$$\begin{array}{rcll} f(0) & = & 0^L & \\ f(1) & = & 1^L & \\ f(x+y) & = & x + y + \bot = x + \bot + y + \bot = f(x) + f(y) & \\ f(xy) & = & xy + \bot & \\ & = & xy + \bot y + x\bot + \bot\bot + \bot & (\text{since } \bot y + x\bot + \bot\bot \leq \bot) \\ & = & (x + \bot)(y + \bot) + \bot & \\ & = & f(x) \cdot^L f(y) & \\ f(\overline{c}) & = & \widetilde{f(c)} \ . & \end{array}$$

It remains to verify $f(x^*) = (f(x))^*$. We have

$$1 + (x + \bot)(x^* + \bot) = 1 + xx^* + x\bot + \bot x^* + \bot\bot \leq x^* + \bot \ ,$$

so the $*$-axioms give us $(x+\bot)^* \leq x^* + \bot$. We have $x^* \leq (x+\bot)^*$ and $\bot \leq (x+\bot)^*$ trivially, so $x^* + \bot \leq (x + \bot)^*$. Therefore, $f(x^*) = x^* + \bot = (x + \bot)^* = (f(x))^*$. So $f : K' \to L$ is a homomorphism.

We now claim that $L \in \mathsf{KAT}$. Since $f : K' \to L$ is a homomorphism and $K' \in \mathsf{KAT}$, L automatically satisfies the equational KAT axioms. We must now verify that L satisfies the two remaining axioms, $p + q \cdot^L x \leq x \to q^* \cdot^L p \leq x$ and $p + x \cdot^L q \leq x \to p \cdot^L q^* \leq x$.

Suppose that $p + q \cdot^L x \leq x$. We must show $q^* \cdot^L p \leq x$. We have $p + qx + \bot = p + q \cdot^L x \leq x$. From $p + qx \leq x$ we conclude $q^* p \leq x$; combining this with $\bot \leq x$, we have $q^* \cdot^L p = q^* p + \bot \leq x$, as desired. Similarly, $p + x \cdot^L q \leq x \to p \cdot^L q^* \leq x$. So $L \in \mathsf{KAT}$.

Define the interpretation $J : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \to L$ by $J(q) = f(I(q))$. Since $K', I \models E$, it immediately follows that $L, J \models E$. Also, $J(r) \leq J(uru) = f(I(uru)) = f(\bot) =$

$\bot + \bot = 0^L$, so $L, J \models r = 0$. Therefore, the assumption $\mathsf{KAT} \models E \wedge r = 0 \rightarrow s = t$ gives us $L, J \models s = t$. Therefore,

$$I(s + uru) = I(s) + I(uru) = I(s) + \bot = f(I(s)) = J(s) = J(t) = I(t + uru) \ .$$

Thus, $K, I \models s + uru = t + uru$. □

Lemma 18

$$\mathsf{KAT}^* \models E \wedge r = 0 \rightarrow s = t \iff \mathsf{KAT}^* \models E \rightarrow s + uru = t + uru$$

Proof The right-to-left implication is as in Lemma 17.

For the left-to-right implication, it suffices to verify that the construction in the proof of Lemma 17 preserves $*$-continuity. Letting $q^{(n)}$ denote the n^{th} power of q under $\cdot^L$ (with $q^{(0)} = 1^L$), we have

$$\begin{aligned}
\sup_n p \cdot^L q^{(n)} \cdot^L r &= \sup_n (pq^n r + \bot) \\
&= pq^* r + \bot \\
&= p \cdot^L q^* \cdot^L r \ .
\end{aligned}$$

□

Lemma 19

$$\mathsf{RKAT} \models E \wedge r = 0 \rightarrow s = t \iff \mathsf{RKAT} \models E \rightarrow s + uru = t + uru$$

Proof The right-to-left implication is as in Lemma 17.

For the left-to-right implication, using the above construction would require verifying that L has a relational representation, which is not clear. Instead, we use a proof-theoretic argument. Suppose $\mathsf{RKAT} \models E \wedge r = 0 \rightarrow \sigma \leq t$, where $\sigma \in R(s)$. $r = 0$ is equivalent to $r \leq 0$, and $\mathsf{KAT} \models t \leq t + uru$, so $\mathsf{RKAT} \models E \wedge r \leq 0 \rightarrow \sigma \leq t + uru$.

For the moment, suppose that the formulas are in the language of KA, so that we can speak about relational proofs without worrying about tests. Let (T, A) be a relational proof of $E \wedge r \leq 0 \rightarrow \sigma \leq t + uru$.

We claim that the hypothesis $r \leq 0$ is never even applied in the proof! Suppose $r \leq 0$ is applied at node $f \in T$ (so f has one child g with $A_g = \text{CON}$). For $r \leq 0$ to be applied at f, there must be states $v, w \in |A_f|$ and $\rho \in R(r)$ with $\rho \in L(A_f^{v,w})$. A property that is preserved in the automata of relational trees is that every state is accessible from the start state a, and the accept state b is accessible from every state. So there exist $\pi \in L(A_f^{a,v})$ and $\pi' \in L(A_f^{w,b})$. Thus, we have $\pi\rho\pi' \in L(A_f)$; we also have $\pi\rho\pi' \in R(uru) \subseteq R(t + uru)$. Therefore, $R(t + uru) \cap L(A_f) \neq \varnothing$, so f is in fact a leaf node, contradicting the assumption that we are applying $r \leq 0$ at

f. (In other words, at any point in a relational tree for $E \wedge r \leq 0 \rightarrow \sigma \leq t + uru$ where we could apply $r \leq 0$, we would already have to be at a leaf.)

So, because $r \leq 0$ is never applied, (T, A) is also a relational proof of $E \rightarrow \sigma \leq t+uru$. Therefore, $\mathsf{RKA} \models E \rightarrow \sigma \leq t+uru$ for all $\sigma \in R(s)$. By Lemma 11, $\mathsf{RKA} \models E \rightarrow s \leq t + uru$, so $\mathsf{RKA} \models E \rightarrow s + uru \leq t + uru$. $\mathsf{RKA} \models E \rightarrow t + uru \leq s + uru$ is similar, and we now have $\mathsf{RKA} \models E \rightarrow s + uru = t + uru$.

In case the formulas are not in the language of KA, we can use the translation from Section 2.5 as follows. We use the above argument to get

$$\mathsf{RKA} \models \mathsf{Tr}(E \wedge r = 0 \rightarrow s = t) \Rightarrow \mathsf{RKA} \models \mathsf{Tr}(E \rightarrow s + uru = t + uru) \ .$$

(The extra hypotheses introduced by the translation may be treated the same as the hypotheses in E. A subtle point here is that the translation introduces new program symbols, without adding them to the universal regular expression; however, the hypotheses added by the tranlation force the interpretations of these extra symbols to be below 1, so they could be added to the universal regular expression without affecting the validity of any formulas involved.) We then have

$$\begin{aligned} \mathsf{RKAT} \models E \wedge r = 0 \rightarrow s = t &\Rightarrow \mathsf{RKA} \models \mathsf{Tr}(E \wedge r = 0 \rightarrow s = t) \\ &\Rightarrow \mathsf{RKA} \models \mathsf{Tr}(E \rightarrow s + uru = t + uru) \\ &\Rightarrow \mathsf{RKAT} \models E \rightarrow s + uru = t + uru \ . \end{aligned}$$

□

Proof of Theorem 16 Immediate from Lemmas 17–19. □

3.2 Idempotent Syntactic Homomorphisms

We can also eliminate hypotheses of the form $cp = c$ (c Boolean, p atomic) in the presence of other hypotheses, but not as cleanly as we eliminated $r = 0$: in this case, the remaining hypotheses will be modified.

The basic idea behind the technique was introduced in [7], which showed how to simultaneously eliminate hypotheses of the form $cp = c$ and $r = 0$. Ernie Cohen later observed that the portion of the proof specific to $cp = c$ was unnecessarily complicated [3]. What we present here is a simplified argument, that is also more general because it works in the presence of other hypotheses. Furthermore, in light of Theorem 16, we no longer need to worry about integrating the elimination of $r = 0$ into the argument, since that can be done separately.

Definition 20 *$H : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \rightarrow \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$ is a* syntactic homomorphism *if for any interpretation $I : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \rightarrow K$ (where $K \in \mathsf{KAT}$), $I \circ H : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \rightarrow K$ is also an interpretation.*

For any syntactic homomorphism $H : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \rightarrow \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$, let E_H be the set of hypotheses

$$\{p = H(p) \ \mid \ p \in \mathsf{P}\} \cup \{b = H(b) \ \mid \ b \in \mathsf{B}\} \ .$$

Definition 20 is equivalent to saying that H is a homomorphism up to KAT-provable equality. A consequence is that H is uniquely determined (up to KAT-provable equality) by its action on P and B; the set of equations E_H then, in a certain sense, captures the action of H.

(For readers familiar with guarded strings, Definition 20 is equivalent to saying that $G \circ H$ is an interpretation, where G is the guarded-string interpretation. More abstractly, the definition is equivalent to saying that H is a lift of an endomorphism on the guarded-string model—that is, there is an endomorphism h on the guarded-string model such that $G \circ H = h \circ G$.)

Lemma 21 *If $H : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \to \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$ is a syntactic homomorphism, then for any $r \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$,*

$$\mathsf{KAT} \models E_H \to r = H(r) \ .$$

Proof Straightforward induction on the structure of r. □

Definition 22 *$H : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \to \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$ is* idempotent *if for all $r \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$,*

$$\mathsf{KAT} \models H(r) = H(H(r)) \ .$$

Theorem 23 *Suppose $H : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \to \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$ is an idempotent syntactic homomorphism, and that E is a set of hypotheses. Let $H(E)$ denote the set of hypotheses*

$$\{H(r) = H(r') \ \mid \ r = r' \text{ is in } E\} \ .$$

Then for any $s, t \in \mathsf{RExp}_{\mathsf{P},\mathsf{B}}$ and $K \in \mathsf{KAT}$,

$$K \models E \land E_H \to s = t \iff K \models H(E) \to H(s) = H(t) \ .$$

Proof For the right-to-left implication, suppose $K \models H(E) \to H(s) = H(t)$ and that we have an intepretation $I : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \to K$ with $K, I \models E \land E_H$. Then by Lemma 21, $K, I \models H(E) \land s = H(s) \land t = H(t)$. It follows by assumption that $K, I \models H(s) = H(t)$. We now have $K, I \models s = H(s) = H(t) = t$. Therefore, $K \models E \land E_H \to s = t$.

For the left-to-right implication, suppose $K \models E \land E_H \to s = t$, and that we have an intepretation $I : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \to K$ with $K, I \models H(E)$. Define $I' : \mathsf{RExp}_{\mathsf{P},\mathsf{B}} \to K$ by $I' = I \circ H$. I' is an interpretation by Definition 20. For any $p \in \mathsf{P}$, idempotence of H gives us $I'(p) = I(H(p)) = I(H(H(p))) = I'(H(p))$; similarly, $I'(b) = I'(H(b))$ for $b \in \mathsf{B}$, so $K, I' \models E_H$. For any equation $r = r'$ in E, $K, I \models H(E)$ gives us $I'(r) = I(H(r)) = I(H(r')) = I'(r')$, so $K, I' \models E$. Therefore, by the assumption $K \models E \land E_H \to s = t$, we have $K, I' \models s = t$, and hence $I(H(s)) = I'(s) = I'(t) = I(H(t))$. Therefore $K, I \models H(s) = H(t)$, as desired. □

Corollary 24 *Suppose F is a set of hypotheses $c_ip_i = c_i$, $1 \le i \le k$, where $p_i \in \mathsf{P}$ are distinct, and each c_i is a Boolean term. Define $H : \mathsf{RExp}_{\mathsf{P,B}} \to \mathsf{RExp}_{\mathsf{P,B}}$ by $H(r) = r[p_i/\overline{c_i}p_i + c_i]$, the result of substituting $\overline{c_i}p_i + c_i$ for p_i in r (for each i). Then for any set E of hypotheses, $s, t \in \mathsf{RExp}_{\mathsf{P,B}}$, and $K \in \mathsf{KAT}$, we have*

$$K \models E \wedge F \to s = t \iff K \models H(E) \to H(s) = H(t) \ .$$

Proof It is easy to verify that H is an idempotent syntactic homomorphism.

Next, observe that $\mathsf{KAT} \models c_ip_i = c_i \leftrightarrow p_i = \overline{c_i}p_i + c_i$. Every equation in E_H is either of the form $p_i = \overline{c_i}p_i + c_i$, or is a tautology such as $b = b$, so F is equivalent to E_H. The corollary now follows immediately from Theorem 23. $\square$

The restriction that the p_i be distinct in Corollary 24 is not a significant imposition, since we can combine $c_ip_i = c_i$ and $c_jp_j = c_j$, for $p_i = p_j$, into $(c_i + c_j)p_i = c_i + c_j$. ($\mathsf{KAT} \models cp = c \wedge dp = d \leftrightarrow (c + d)p = c + d$ is easy to verify: $\to$ is trivial, and $\leftarrow$ follows by multiplying $(c + d)p = c + d$ on the left by c or d.)

4 Acknowledgments

This work was supported in part by NSF grant CCR-0105586 and by ONR Grant N00014-01-1-0968. The views and conclusions contained herein are those of the author 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] 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.

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

[3] Ernie Cohen, 2003. Private communication.

[4] 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.

[5] Chris Hardin. *The Horn Theory of Relational Kleene Algebra*. PhD thesis, Cornell University, 2005. In preparation.

[6] Chris Hardin. Proof theory for Kleene algebra. In P. Panangaden and ???, editors, *Proc. 20th Symposium on Logic in Computer Science (LICS 2005)*, volume ??? of *???*, pages ???–???, Chicago, June 2005. IEEE.

[7] Chris Hardin and Dexter Kozen. On the elimination of hypotheses in Kleene algebra with tests. Technical Report 2002-1879, Computer Science Department, Cornell University, October 2002.

[8] Chris Hardin and Dexter Kozen. On the complexity of the Horn theory of REL. Technical Report 2003-1896, Computer Science Department, Cornell University, May 2003.

[9] Dexter Kozen. *The Design and Analysis of Algorithms*. Springer-Verlag, New York, 1991.

[10] Dexter Kozen. Kleene algebra with tests. *Transactions on Programming Languages and Systems*, pages 427–443, 1997.

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

[12] Dexter Kozen. On the complexity of reasoning in Kleene algebra. *Information and Computation*, 179:159–162, 2002.

[13] Dexter Kozen and Maria-Cristina Patron. Certification of compiler optimizations using Kleene algebra with tests. In J. Lloyd, V. Dahl, U. Furbach, M. Kerber, K.-K. Lau, C. Palamidessi, L. M. Pereira, Y. Sagiv, and P. 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.

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