Substructural Logic and Partial Correctness

DEXTER KOZEN
Cornell University

and
JERZY TIURYN
Warsaw University

We formulate a noncommutative sequent calculus for partial correctness that subsumes propositional Hoare Logic. Partial correctness assertions are represented by intuitionistic linear implication. We prove soundness and completeness over relational and trace models. As a corollary we obtain a complete sequent calculus for inclusion and equivalence of regular expressions.

Categories and Subject Descriptors: D.2.2 [**Software Engineering**]: Tools and Techniques—*structured programming*; D.2.4 [**Software Engineering**]: Program Verification—*correctness proofs*; D.3.3 [**Software Engineering**]: Language Constructs and Features—*control structures*; F.3.1 [**Logics and Meanings of Programs**]: Specifying and Verifying and Reasoning about Programs—*assertions*; *invariants*; *logics of programs*; *mechanical verification*; *pre- and postconditions*; *specification techniques*; F.3.2 [**Logics and Meanings of Programs**]: Semantics of Programming Languages—*algebraic approaches to semantics*; F.3.3 [**Logics and Meanings of Programs**]: Studies of Program Constructs—*control primitives*; I.1.1 [**Algebraic Manipulation**]: Expressions and Their Representations—*simplification of expressions*; I.1.3 [**Algebraic Manipulation**]: Languages and Systems—*special-purpose algebraic systems*; I.2.2 [**Algebraic Manipulation**]: Automatic Programming—*program modification*; *program synthesis*; *program transformation*; *program verification*

General Terms: Theory; Verification

Additional Key Words and Phrases: Dynamic logic, Hoare logic, Kleene algebra, Kleene algebra with tests, linear logic, sequent calculus, specification, substructural logic

1. INTRODUCTION

In formulating logics for program verification such as Hoare Logic (HL), Dynamic Logic (DL), or Kleene Algebra with Tests (KAT), it is tempting to treat tests and correctness assertions as a uniform syntactic category. This temptation is best resisted: although both are classes of assertions, they have quite different characteristics. *Tests* are local assertions whose truth is determined by the current state of execution. They are normally immediately decidable. The assertion $x \geq 0$,

Supported in part by NSF grant CCR-0105586, ONR grant N00014-01-1-0968, and Polish KBN Grant 7 T11C 028 20. The views and conclusions contained herein are those of the authors and do not necessarily represent the official policies or endorsements, either expressed or implied, of these organizations or the US Government.

where x is a program variable, is an example of such a test. Tests occur in all modern programming languages as part of conditional expressions and looping constructs. *Correctness assertions*, on the other hand, are statements about the global behavior of a program, such as partial correctness or halting. They are typically much richer in expressive power than tests and undecidable in general.

DL does not distinguish between these two categories of assertions. The two are freely mixed, and both are treated classically. For this reason, the resulting system is unnecessarily complex for its purposes. The rich-test version of DL, in which one can convert an arbitrary correctness assertion to a test using the operator ?, is Π^1_1-complete (see [Harel et al. 2000]). Even with systems that do make the distinction, such as KAT, care must be taken not to inadvertently treat global properties as local; doing so can lead to anomalies such as the Dead Variable Paradox [Kozen and Patron 2000].

One major distinguishing factor between tests and correctness assertions that may not be immediately apparent is that the former are classical in nature, whereas the latter are intuitionistic. For example, the DL axiom

$$[p][q]b \;\equiv\; [p\,;\,q]b$$

can be regarded as a noncommutative version of the intuitionistic currying rule

$$p \to q \to b \;\equiv\; p \wedge q \to b.$$

Gödel [1933] first observed the strong connection between modal and intuitionistic logic, foreshadowing Kripke's [1963; 1965] formulation of similar state-based semantics for these logics (see [Artemov 2001]). Kripke models also form the basis of the standard semantics of DL (see [Harel et al. 2000]), although as mentioned, DL does not realize the intuitionistic nature of partial correctness.

In this paper we give a sequent calculus S that clearly separates partial correctness reasoning into its classical and intuitionistic parts. The system can be viewed as a *substructural logic*. These logics result from restricting the structural rules (weakening, exchange, contraction) in various ways. The interested reader is referred to [Restall 2000] for a thorough introduction to substructural logics. We will explain later how some of the structural rules of the present system are restricted. In Section 4, where we introduce the system, we will explain why we view partial correctness reasoning in S as intuitionistic rather than classical.

The system has two syntactic categories: *programs* and *formulas*. *Tests* comprise the intersection of these two categories. Tests are boolean combinations of propositional variables. Reasoning about tests uses classical logic.

Programs are represented by regular expressions. They are formed from atomic programs and tests with help of composition $\otimes$, nondeterministic choice $\oplus$, and iteration $^+$. The notation for the program connectives $\otimes$ and $\oplus$ is chosen to stress their relationship to the well known linear logic connectives: multiplicative conjunction $\otimes$ and additive disjunction $\oplus$ (see [Girard 1987]). As could be expected, $\otimes$ in our formalism is noncommutative.

Formulas are built from tests and programs using implication. Intuitively, formulas represent weakest preconditions or box formulas of DL. There is a syntactic restriction: an implication $p \to \varphi$ may only be formed from a program p on the left and a formula φ on the right. Hence the general form of a formula is

$p_1 \to p_2 \to \cdots \to p_n \to c$, where $p_1, \ldots, p_n$ are programs and c is a test. Because of this restriction and because of the severe restrictions on structural rules regarding programs, implication has a linear flavor. Also, due to the form of the rules of inference for handling implication, it follows that implication is intuitionistic.

Sequents of the system are of the form $\Gamma \vdash \varphi$, where Γ is a sequence of programs and formulas and φ is a formula. The sequence Γ is called an *environment*. Programs and formulas are treated differently, as can be seen from the structural rules. There is a weakening rule for formulas, but only a very restricted weakening for programs: they can be inserted only in front of the environment. The contraction rule, although absent in the system, is derivable for formulas, but it is not derivable for programs. There is no exchange rule, although some weak forms of it can be derived. There is a co-contraction rule: a program of the form p^+ already present in the environment can be duplicated. Troelstra [1992, p. 25] remarks that contraction has more dramatic proof-theoretic consequences than weakening when added to Linear Logic.

The system has introduction rules for implication on the left and on the right of $\vdash$. Due to the asymmetrical structure of sequents, each of the program connectives has introduction and elimination rules exclusively on the left side of $\vdash$. In this sense, the system is neither in the style of natural deduction (introduction/elimination on the right), nor in the style of the Gentzen calculus (introduction on the left and on the right). As mentioned earlier, the system has three structural rules and a cut rule.

The paper is organized as follows. In Section 2 we introduce the syntax of the language of System S. In Section 3 we give relational and trace semantics for this logic and show how the logic captures partial correctness. In Section 4, which is the main technical part of the paper, we introduce the rules of System S and establish its basic properties needed later in the proof of the completeness result. The completeness proof relies on results from Kleene algebra. A relationship between System S and Kleene algebra, together with some properties used in the proof of completeness, are presented in Section 4.2. As a corollary (Proposition 4.13), we obtain a complete sequent calculus for inclusion and equivalence of regular expressions. In Section 4.3 we show two examples of valid rules for reasoning about partial correctness assertions which are not derivable in Hoare logic but are derivable in System S. Section 5 is devoted to the soundness of S and Section 6 to its completeness over both classes of models.

We mention that our two equivalent semantics of Section 3 are both special cases of a more general approach to the semantics of noncommutative Linear Logic via quantales [Yetter 1990]. We restrict our attention to two special kinds of quantales: sets of traces and binary relations. Our completeness result is thus stronger than it would be for the more general semantics based on arbitrary quantales.

2. SYNTAX

The syntax of S comprises several syntactic categories. These will require some intuitive explanation, which we defer until after the formal definition. In particular

we distinguish between two kinds of propositions, which we call *tests* and *formulas*.

tests	$b, c, d \ldots$	$b ::= \langle \text{atomic tests} \rangle \mid \mathbf{0} \mid b \to c$
programs	$p, q, r, \ldots$	$p ::= \langle \text{atomic programs} \rangle \mid b \mid p \oplus q \mid p \otimes q \mid p^+$
formulas	$\varphi, \psi, \ldots$	$\varphi ::= b \mid p \to \varphi$
environments	$\Gamma, \Delta, \ldots$	$\Gamma ::= \varepsilon \mid \Gamma, p \mid \Gamma, \varphi$
sequents	$\Gamma \vdash \varphi$	

In the above grammar, $\to$ is called *linear implication*, $\otimes$ is a noncommutative multiplicative connective called *tensor*, $\oplus$ is a commutative additive connective called *disjunction*, and $^+$ is a unary operation called *positive iteration*. We use brackets where necessary to ensure unique readability. We abbreviate $b \to \mathbf{0}$ by $\overline{b}$, $\overline{\mathbf{0}}$ by $\mathbf{1}$, $p \otimes q$ by pq, and $\mathbf{1} \oplus p^+$ by p^*.

Several formalisms, such as PDL [Fischer and Ladner 1979] and KAT [Kozen 1997], are based on * rather than $^+$. We can freely move between the two languages since * and $^+$ are mutually definable:

$$p^* \;=\; \mathbf{1} \oplus p^+ \qquad p^+ \;=\; pp^*.$$

For this reason, models for one language can be viewed as models for the other.

We base S on $^+$ instead of * because the resulting deductive system is cleaner—it contains no contraction rule[1]. This is perhaps due to the fact that $^+$ can be viewed as a more primitive operation than *.

A *test* is either an atomic test, the symbol $\mathbf{0}$ representing falsity, or an expression $b \to c$ representing classical implication, where b and c are tests. We use the symbols $b, c, d, \ldots$ exclusively to stand for tests. The set of all tests is denoted $\mathcal{B}$. The sequent calculus to be presented in Section 4 will encode classical propositional logic for tests.

A *program* is either an atomic program, a test, or an expression $p \oplus q$, $p \otimes q$, or p^+, where p and q are programs. We use the symbols $p, q, r, \ldots$ exclusively to stand for programs. The set of all programs is denoted $\mathcal{P}$. As in PDL [Fischer and Ladner 1979], the program operators can be used to construct conventional procedural programming constructs such as conditional tests and while loops.

A *formula* is either a test or an expression $p \to \varphi$, read "after p, φ," where p is a program and φ is a formula. Intuitively, the meaning is similar to the DL modal construct $[p]\varphi$. The operator $\to$ associates to the right. We use the symbols $\varphi, \psi, \ldots$ to stand for formulas.

Environments are denoted $\Gamma, \Delta, \ldots$. An environment is a (possibly empty) sequence of programs and formulas. The empty environment is denoted ε. Intuitively, an environment describes a previous computation that has led to the current state.

Sequents are of the form $\Gamma \vdash \varphi$, where Γ is an environment and φ is a formula. We write $\vdash \varphi$ for $\varepsilon \vdash \varphi$. Intuitively, the meaning of $\Gamma \vdash \varphi$ is similar to the DL assertion $[\Gamma]\varphi$, where we think of the environment $\Gamma = \ldots, p, \ldots, \psi, \ldots$ as the rich-test program $\cdots ; p; \cdots ; \psi?; \cdots$ of DL.

[1] In fact, one of the natural rules for * is a co-weakening rule, which is a strong form of a contraction rule.

3. SEMANTICS

3.1 Guarded Strings

Guarded strings over P, B were introduced in [Kaplan 1969] (see also [Kozen and Smith 1996]). We review the definition here.

Let $\mathsf{B} = \{b_1, \ldots, b_k\}$ and $\mathsf{P} = \{p_1, \ldots, p_m\}$ be fixed disjoint finite sets of atomic tests and atomic programs, respectively. An *atom* of B is a program $\ell_1 \cdots \ell_k$ such that ℓ_i is either b_i or $\overline{b}_i$. We require for technical reasons that the ℓ_i occur in this order. An atom represents a minimal nonzero element of the free Boolean algebra on B. We denote by $\mathcal{A}_\mathsf{B}$ the set of all atoms of B. For an atom α and a test b, we write $\alpha \leq b$ if $\alpha \to b$ is a classical propositional tautology.

A *guarded string* is a sequence

$$\sigma \;=\; \alpha_0 q_0 \alpha_1 \cdots \alpha_{n-1} q_{n-1} \alpha_n,$$

where $n \geq 0$, each $\alpha_i \in \mathcal{A}_\mathsf{B}$, and $q_i \in \mathsf{P}$. We define $\mathbf{first}(\sigma) = \alpha_0$ and $\mathbf{last}(\sigma) = \alpha_n$.

If $\mathbf{last}(\sigma) = \mathbf{first}(\tau)$, we can form the *fusion product* $\sigma\tau$ by concatenating σ and τ, omitting the extra copy of $\mathbf{last}(\sigma) = \mathbf{first}(\tau)$ in between. For example, if $\sigma = \alpha p \beta$ and $\tau = \beta q \gamma$, then $\sigma\tau = \alpha p \beta q \gamma$. If $\mathbf{last}(\sigma) \neq \mathbf{first}(\tau)$, then $\sigma\tau$ does not exist. The notation $\sigma\tau$ for fusion product should not be misinterpreted as concatenation of strings; the latter operation is not defined for guarded strings.

For sets X, Y of guarded strings, define

$$X \circ Y \;\stackrel{\text{def}}{=}\; \{\sigma\tau \mid \sigma \in X,\ \tau \in Y,\ \mathbf{last}(\sigma) = \mathbf{first}(\tau)\}$$
$$X^0 \;\stackrel{\text{def}}{=}\; \mathcal{A}_\mathsf{B}, \qquad X^{n+1} \;\stackrel{\text{def}}{=}\; X \circ X^n.$$

Although fusion product is a partial operation on guarded strings, the operation $\circ$ is a total operation on sets of guarded strings. If there is no existing fusion product between an element of X and an element of Y, then $X \circ Y = \varnothing$.

Each program p denotes a set $GS(p)$ of guarded strings:

$$\begin{aligned}
GS(p) &\stackrel{\text{def}}{=} \{\alpha p \beta \mid \alpha, \beta \in \mathcal{A}_\mathsf{B}\}, \quad p \text{ atomic} \\
GS(b) &\stackrel{\text{def}}{=} \{\alpha \in \mathcal{A}_\mathsf{B} \mid \alpha \leq b\}, \quad b \text{ a test} \\
GS(p \oplus q) &\stackrel{\text{def}}{=} GS(p) \cup GS(q) \\
GS(p \otimes q) &\stackrel{\text{def}}{=} GS(p) \circ GS(q) \\
GS(p^+) &\stackrel{\text{def}}{=} \bigcup_{n \geq 1} GS(p)^n.
\end{aligned}$$

It follows that $GS(p^*) = \bigcup_{n \geq 0} GS(p)^n$. A guarded string σ is itself a program, and $GS(\sigma) = \{\sigma\}$.

A set of guarded strings over P, B is *regular* if it is $GS(p)$ for some program p. The regular sets of guarded strings form the free Kleene algebra with tests on generators P, B [Kozen and Smith 1996]; in other words, $GS(p) = GS(q)$ iff $p = q$ is a theorem of KAT.

LEMMA 3.1. *The regular sets of guarded strings are closed under the Boolean operations.*

PROOF. Closure under $\varnothing$ and union are explicit by means of the constructs $\mathbf{0}$ and $\oplus$. It was shown in [Kaplan 1969, Theorem 5] (see also [Kozen and Smith 1996]) that for any program p, there is an equivalent program $\widehat{p}$ such that $GS(p) = GS(\widehat{p}) = R(\widehat{p})$, where $R(\widehat{p})$ is the regular set of strings over the alphabet $\mathsf{P} \cup \mathsf{B} \cup \{\overline{b} \mid b \in \mathsf{B}\}$ denoted by $\widehat{p}$ under the usual interpretation of regular expressions. For example, if $w = (p_1 \oplus \cdots \oplus p_m)^*$, we might take $\widehat{w} = (b(p_1 \oplus \cdots \oplus p_m))^* b$, where $b = (b_1 \oplus \overline{b}_1) \cdots (b_k \oplus \overline{b}_k)$. The set $GS(w) = GS(\widehat{w}) = R(\widehat{w})$ is the set of all guarded strings.

It remains to show closure under complement; closure under intersection follows by the De Morgan laws. Let p' be an expression such that $R(p') = R(\widehat{w}) - R(\widehat{p})$. The expression p' exists since the regular sets of strings over $\mathsf{P} \cup \mathsf{B} \cup \{\overline{b} \mid b \in \mathsf{B}\}$ are closed under the Boolean operations. Then $R(p')$ is a set of guarded strings since $R(\widehat{w})$ is, and

$$GS(p') = R(p') = R(\widehat{w}) - R(\widehat{p}) = GS(w) - GS(p).$$

□

3.2 Trace Models

Traces are similar to guarded strings but more general. They are defined in terms of Kripke frames. A *Kripke frame* over P, B is a structure $(K, \mathfrak{m}_K)$, where

$$\mathfrak{m}_K : \mathsf{P} \to 2^{K \times K} \qquad \mathfrak{m}_K : \mathsf{B} \to 2^K.$$

Although syntactically every test is a program, the primitive test symbols B and primitive program symbols P are disjoint sets, so there is no ambiguity in the definition of $\mathfrak{m}_K$.

Elements of K are called *states*. A *trace* in K is an alternating sequence of states and primitive program symbols $s_0 q_0 s_1 \cdots s_{n-1} q_{n-1} s_n$, where $n \geq 0$, $s_i \in K$, $q_i \in \mathsf{P}$, and $(s_i, s_{i+1}) \in \mathfrak{m}_K(q_i)$ for $0 \leq i \leq n-1$. The first and last states of 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 $\sigma\tau$. If $\mathbf{last}(\sigma) \neq \mathbf{first}(\tau)$ then $\sigma\tau$ does not exist. A trace $s_0 q_0 s_1 \cdots s_{n-1} q_{n-1} s_n$ is *acyclic* if the s_i are distinct. The model K is *acyclic* if all traces are acyclic. It is no loss of generality to restrict attention to acyclic models; every model is equivalent to an acyclic model obtained by "unwinding" the original model (see [Harel et al. 2000, p. 132] for an explicit construction).

If X and Y are sets of traces, define

$$X \circ Y \stackrel{\text{def}}{=} \{\sigma\tau \mid \sigma \in X,\ \tau \in Y,\ \mathbf{last}(\sigma) = \mathbf{first}(\tau)\}$$
$$X^0 \stackrel{\text{def}}{=} K, \qquad X^{n+1} \stackrel{\text{def}}{=} X \circ X^n.$$

Tests, programs, formulas, and environments are interpreted as sets of traces ac-

cording to the following inductive definition:

$$\begin{aligned}
[\![p]\!]_K &\stackrel{\text{def}}{=} \{spt \mid (s,t) \in \mathfrak{m}_K(p)\}, \quad p \text{ atomic}\\
[\![b]\!]_K &\stackrel{\text{def}}{=} \mathfrak{m}_K(b), \quad b \text{ atomic}\\
[\![\mathbf{0}]\!]_K &\stackrel{\text{def}}{=} \varnothing\\
[\![p \oplus q]\!]_K &\stackrel{\text{def}}{=} [\![p]\!]_K \cup [\![q]\!]_K\\
[\![p \otimes q]\!]_K &\stackrel{\text{def}}{=} [\![p]\!]_K \circ [\![q]\!]_K\\
[\![p^+]\!]_K &\stackrel{\text{def}}{=} \bigcup_{n\geq 1} [\![p]\!]_K^n\\
[\![p \to \varphi]\!]_K &\stackrel{\text{def}}{=} \{s \mid \forall \tau\ \mathbf{first}(\tau) = s \text{ and } \tau \in [\![p]\!]_K \ \Rightarrow\ \mathbf{last}(\tau) \in [\![\varphi]\!]_K\}\\
[\![\varepsilon]\!]_K &\stackrel{\text{def}}{=} K\\
[\![\Gamma, \Delta]\!]_K &\stackrel{\text{def}}{=} [\![\Gamma]\!]_K \circ [\![\Delta]\!]_K.
\end{aligned}$$

It follows that

$$\begin{aligned}
[\![\overline{b}]\!]_K &= K - [\![b]\!]_K\\
[\![\mathbf{1}]\!]_K &= K\\
[\![p^*]\!]_K &= \bigcup_{n\geq 0} [\![p]\!]_K^n.
\end{aligned}$$

Every trace σ has an associated guarded string $\mathrm{gs}(\sigma)$ defined by

$$\mathrm{gs}(s_0 q_0 s_1 \cdots s_{n-1} q_{n-1} s_n) \stackrel{\text{def}}{=} \alpha_0 q_0 \alpha_1 \cdots \alpha_{n-1} q_{n-1} \alpha_n,$$

where α_i is the unique atom of B such that $s_i \in [\![\alpha_i]\!]_K$. The atom α_i is unique because for each $b \in \mathsf{B}$, exactly one of $s_i \in [\![b]\!]_K$ or $s_i \in [\![\overline{b}]\!]_K$. Thus $\mathrm{gs}(\sigma)$ is the unique guarded string over P, B such that $\sigma \in [\![\mathrm{gs}(\sigma)]\!]_K$. The guarded string $\mathrm{gs}(\sigma)$ is unique, because for any guarded string $\beta_0 p_0 \beta_1 \cdots \beta_{m-1} p_{m-1} \beta_m$, any trace in $[\![\beta_0 p_0 \beta_1 \cdots \beta_{m-1} p_{m-1} \beta_m]\!]_K$ must be of the form $s_0 p_0 s_1 \cdots s_{m-1} p_{m-1} s_m$ such that $s_i \in [\![\beta_i]\!]_K$, $0 \leq i \leq m$.

The sequent $\Gamma \vdash \varphi$ is *valid* in the trace model K if for all traces $\sigma \in [\![\Gamma]\!]_K$, $\mathbf{last}(\sigma) \in [\![\varphi]\!]_K$; equivalently, if $[\![\Gamma]\!]_K \subseteq [\![\Gamma, \varphi]\!]_K$. A sequent is *valid* if it is valid in all trace models over P and B.

Guarded strings (Section 3.1) are just traces of a Kripke frame G whose states are atoms of B with

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

In the notation of this section, $GS(p)$ would be denoted $[\![p]\!]_G$.

The relationship between trace semantics and guarded strings is given by the following lemma.

LEMMA 3.2. *In any trace model K, for any program p and trace τ, $\tau \in [\![p]\!]_K$ iff $\mathrm{gs}(\tau) \in GS(p)$. In other words, $[\![p]\!]_K = \mathrm{gs}^{-1}(GS(p))$. The map $X \mapsto \mathrm{gs}^{-1}(X)$ is a* KAT *homomorphism from the algebra of regular sets of guarded strings to the algebra of regular sets of traces over K.*

PROOF. Induction on the structure of p. □

3.3 Relational Models

Kripke frames $(K,\ \mathfrak{m}_K)$ also give rise to relational models. In a relational model, tests, programs, formulas, and environments are interpreted as binary relations on K. Tests and formulas denote subsets of the identity relation.

$$\begin{aligned}
[\![p]\!]_K &\stackrel{\text{def}}{=} \mathfrak{m}_K(p), \quad p \text{ atomic}\\
[\![b]\!]_K &\stackrel{\text{def}}{=} \{(s,s) \mid s \in \mathfrak{m}_K(b)\}, \quad b \text{ atomic}\\
[\![\mathbf{0}]\!]_K &\stackrel{\text{def}}{=} \varnothing\\
[\![p \oplus q]\!]_K &\stackrel{\text{def}}{=} [\![p]\!]_K \cup [\![q]\!]_K\\
[\![p \otimes q]\!]_K &\stackrel{\text{def}}{=} [\![p]\!]_K \circ [\![q]\!]_K\\
[\![p^+]\!]_K &\stackrel{\text{def}}{=} \bigcup_{n\geq 1} [\![p]\!]_K^n\\
[\![p \to \varphi]\!]_K &\stackrel{\text{def}}{=} \{(s,s) \mid \forall t\ (s,t) \in [\![p]\!]_K \ \Rightarrow\ (t,t) \in [\![\varphi]\!]_K\}\\
[\![\varepsilon]\!]_K &\stackrel{\text{def}}{=} \{(s,s) \mid s \in K\}\\
[\![\Gamma, \Delta]\!]_K &\stackrel{\text{def}}{=} [\![\Gamma]\!]_K \circ [\![\Delta]\!]_K,
\end{aligned}$$

where $\circ$ denotes ordinary relational composition. It follows that

$$\begin{aligned}
[\![\overline{b}]\!]_K &= \{(s,s) \mid (s,s) \notin [\![b]\!]_K\}\\
[\![\mathbf{1}]\!]_K &= \{(s,s) \mid s \in K\}\\
[\![p^*]\!]_K &= \bigcup_{n\geq 0} [\![p]\!]_K^n.
\end{aligned}$$

Writing $s \vDash \varphi$ for $(s,s) \in [\![\varphi]\!]_K$, the defining clause for $p \to \varphi$ becomes

$$s \vDash p \to \varphi \ \Leftrightarrow\ \forall t\ (s,t) \in [\![p]\!]_K \Rightarrow t \vDash \varphi,$$

thus the meaning of $p \to \varphi$ is essentially the same as the meaning of the box formula $[\![p]\!]\varphi$ of DL.

The sequent $\Gamma \vdash \varphi$ is *valid* in the relational model on $(K, \mathfrak{m}_K)$ if for all $s,t \in K$, if $(s,t) \in [\![\Gamma]\!]_K$, then $(t,t) \in [\![\varphi]\!]_K$; equivalently, if the DL formula $[\![\Gamma]\!]\varphi$ is true in all states under the rich-test semantics [Fischer and Ladner 1979], where the environment $\Gamma = \ldots, p, \ldots, \psi, \ldots$ is interpreted as the rich-test program $\cdots\,; p; \cdots\,; \psi?; \cdots$.

3.4 Relationship between Trace and Relational Models

The following theorem and corollary establish the connection with the standard relational semantics of DL.

THEOREM 3.3. *The map*

$$\mathrm{Ext} : X \ \mapsto\ \{(\mathbf{first}(\sigma), \mathbf{last}(\sigma)) \mid \sigma \in X\}$$

taking sets of traces on K to binary relations on K commutes with $[\![\,]\!]_K$ and $[\![\,]\!]_K$; that is, $\mathrm{Ext}([\![E]\!]_K) = [\![E]\!]_K$ *for any test, program, formula, or environment E.*

PROOF. The proof is a straightforward induction on syntax, using the fact that Ext commutes with the operators $\cup$ and $\circ$ on sets of traces and binary relations. □

COROLLARY 3.4. *Validity over relational models is the same as validity over trace models.*

PROOF. Suppose that $\Gamma \vdash \varphi$ is valid in the trace model over K. If $(s,t) \in [\![\Gamma]\!]_K$, then there exists a trace $\tau \in [\![\![\Gamma]\!]\!]_K$ such that $s = \mathbf{first}(\tau)$ and $t = \mathbf{last}(\tau)$. By the assumption, $[\![\![\Gamma]\!]\!]_K \subseteq [\![\![\Gamma, \varphi]\!]\!]_K$, thus $t \in [\![\![\varphi]\!]\!]_K$ and $(t,t) \in [\![\varphi]\!]_K$. This says that $\Gamma \vdash \varphi$ is valid in the relational model over K.

Conversely, suppose that $\Gamma \vdash \varphi$ is valid in the relational model over K. If $\tau \in [\![\![\Gamma]\!]\!]_K$, then $(\mathbf{first}(\tau), \mathbf{last}(\tau)) \in [\![\Gamma]\!]_K$. By the assumption, $(\mathbf{last}(\tau), \mathbf{last}(\tau)) \in [\![\varphi]\!]_K$, thus $\mathbf{last}(\tau) \in [\![\![\varphi]\!]\!]_K$. This says that $\Gamma \vdash \varphi$ is valid in the trace model over K. □

4. A DEDUCTIVE SYSTEM

The rules of System S are given in Figure 1. All rules are of the form

$$\frac{\Gamma_1 \vdash \varphi_1 \quad \ldots \quad \Gamma_n \vdash \varphi_n}{\Gamma \vdash \varphi} \ .$$

The sequents above the line are the *premises* and the sequent below the line is the *conclusion*. Since programs cannot occur positively on the right hand side of $\vdash$, the system has introduction and elimination rules on the left of $\vdash$.

We will use the notation $\Gamma \vdash \varphi$ ambiguously as both an object and a meta-assertion. As an object it denotes a sequent, *i.e.* a sequence of symbols over the appropriate vocabulary. As a meta-assertion it says that the sequent $\Gamma \vdash \varphi$ is provable in S. In particular, $\Gamma \nvdash \varphi$ means that the sequent $\Gamma \vdash \varphi$ is not provable in S. The proper interpretation should always be clear from context.

Let us briefly explain some of the rules of S. The rule **(test-cut)** implies the classical nature of tests. The rule $(\mathbf{I}^+)$ says that if p is partially correct with respect to the precondition ψ and the postcondition φ, and if ψ is an invariant for p, then the iteration p^+ is also partially correct with respect to the same pre- and postconditions. So this is clearly related to a standard rule of Hoare logic. Recall that, as far as relational models are concerned, validity of a sequent $\Gamma \vdash \varphi$ means that for every pair of states s, t, if Γ transforms s to t, then t must satisfy φ. Since formulas that occur as elements of Γ act as filters (*i.e.*, as partial identities on states that satisfy them), it follows that we can always insert a formula in any place in the environment, since it can only reduce the number of reachable states, thereby not affecting the validity of the sequent. Thus the full weakening rule $(\mathbf{W}\ \psi)$ for formulas is sound. On the other hand, for the same reason, a program can be inserted soundly only at the begining of the environment. This explains the difference between $(\mathbf{W}\ \psi)$ and $(\mathbf{W}\ p)$.

A rule is *admissible* if for any substitution instance for which the premises are provable, the conclusion is also provable. The proof of the conclusion may depend on the structure of the expressions substituted for the metasymbols appearing in the rule or on the proofs of the premises. To show admissibility, it suffices to derive the conclusion in S augmented with the premises as extra axioms, considering the metasymbols appearing in the rule as atomic symbols in the object language. Any

Axiom (b is a test):

$$b \vdash b$$

Test-cut Rule (b is a test):

$$\textbf{(test-cut)} \quad \frac{\Gamma, b, \Delta \vdash \varphi \qquad \Gamma, \overline{b}, \Delta \vdash \varphi}{\Gamma, \Delta \vdash \varphi}$$

Arrow Rules:

$$\textbf{(R} \rightarrow\textbf{)} \quad \frac{\Gamma, p \vdash \varphi}{\Gamma \vdash p \rightarrow \varphi}$$

$$\textbf{(I} \rightarrow\textbf{)} \quad \frac{\Gamma, p, \psi, \Delta \vdash \varphi}{\Gamma, p \rightarrow \psi, p, \Delta \vdash \varphi}$$

Introduction Rules:

$$\textbf{(I} \otimes\textbf{)} \quad \frac{\Gamma, p, q, \Delta \vdash \varphi}{\Gamma, p \otimes q, \Delta \vdash \varphi}$$

$$\textbf{(I} \oplus\textbf{)} \quad \frac{\Gamma, p, \Delta \vdash \varphi \qquad \Gamma, q, \Delta \vdash \varphi}{\Gamma, p \oplus q, \Delta \vdash \varphi}$$

$$\textbf{(I 0)} \quad \Gamma, \mathbf{0}, \Delta \vdash \varphi$$

$$\textbf{(I}^{+}\textbf{)} \quad \frac{\psi, p \vdash \varphi \qquad \psi, p \vdash \psi}{\psi, p^{+} \vdash \varphi}$$

Elimination Rules:

$$\textbf{(E} \otimes\textbf{)} \quad \frac{\Gamma, p \otimes q, \Delta \vdash \varphi}{\Gamma, p, q, \Delta \vdash \varphi}$$

$$\textbf{(E1} \oplus\textbf{)} \quad \frac{\Gamma, p \oplus q, \Delta \vdash \varphi}{\Gamma, p, \Delta \vdash \varphi}$$

$$\textbf{(E2} \oplus\textbf{)} \quad \frac{\Gamma, p \oplus q, \Delta \vdash \varphi}{\Gamma, q, \Delta \vdash \varphi}$$

$$\textbf{(E}^{+}\textbf{)} \quad \frac{\Gamma, p^{+}, \Delta \vdash \varphi}{\Gamma, p, \Delta \vdash \varphi}$$

Structural Rules:

$$\textbf{(W} \psi\textbf{)} \quad \frac{\Gamma, \Delta \vdash \varphi}{\Gamma, \psi, \Delta \vdash \varphi}$$

$$\textbf{(W} p\textbf{)} \quad \frac{\Gamma \vdash \varphi}{p, \Gamma \vdash \varphi}$$

$$\textbf{(CC}^{+}\textbf{)} \quad \frac{\Gamma, p^{+}, \Delta \vdash \varphi}{\Gamma, p^{+}, p^{+}, \Delta \vdash \varphi}$$

Cut Rule:

$$\textbf{(cut)} \quad \frac{\Gamma \vdash \psi \qquad \Gamma, \psi, \Delta \vdash \varphi}{\Gamma, \Delta \vdash \varphi}$$

Fig. 1. Rules of System S

such derivation will then be uniformly valid over all substitution instances. For example, the following *contraction rule*

$$\textbf{(C} \psi\textbf{)} \quad \frac{\Gamma, \psi, \psi, \Delta \vdash \psi}{\Gamma, \psi, \Delta \vdash \psi}$$

is admissible in S. Indeed, below is a derivation of the conclusion from the premise of **(C** ψ**)**.

$$\dfrac{\begin{array}{c}\psi \vdash \psi \\ \vdots \ \textbf{(W} p\textbf{), (W} \psi\textbf{)} \\ \Gamma, \psi \vdash \psi \end{array} \qquad \Gamma, \psi, \psi, \Delta \vdash \varphi}{\Gamma, \psi, \Delta \vdash \psi}\ \textbf{(cut)}$$

where the sequent $\psi \vdash \psi$ is derivable by Lemma 4.2. Note that the above derivation depends on Γ. Also, as can be seen from the proof Lemma 4.2, the derivation of $\psi \vdash \psi$ depends on ψ.

4.1 Basic Properties

LEMMA 4.1. *The rule*

$$\textbf{(E 1)} \quad \frac{\Gamma, \mathbf{1}, \Delta \vdash \varphi}{\Gamma, \Delta \vdash \varphi}$$

is admissible.

PROOF. We have the following derivation. Note that $\Gamma, \mathbf{0}, \Delta \vdash \varphi$ is an instance of **(I 0)**.

$$\frac{\Gamma, \mathbf{1}, \Delta \vdash \varphi \quad \Gamma, \mathbf{0}, \Delta \vdash \varphi}{\Gamma, \Delta \vdash \varphi}\ \textbf{(test-cut)}$$

□

LEMMA 4.2. *The rule and sequent*

$$\textbf{(mono)} \quad \frac{\varphi \vdash \psi}{p \to \varphi \vdash p \to \psi} \qquad \textbf{(ident)} \quad \varphi \vdash \varphi$$

are admissible.

PROOF. The following diagram gives a proof of **(mono)**.

$$\cfrac{\cfrac{\cfrac{\varphi \vdash \psi}{p, \varphi \vdash \psi}\ \textbf{(W}\ p\textbf{)}}{p \to \varphi, p \vdash \psi}\ \textbf{(I}\to\textbf{)}}{p \to \varphi \vdash p \to \psi}\ \textbf{(R}\to\textbf{)}$$

The identity sequent **(ident)** follows by induction on the structure of φ using **(mono)**. The basis $b \vdash b$ is an instance of the axiom. □

LEMMA 4.3. *The rules*

$$\textbf{(ER}\to\textbf{)} \quad \frac{\Gamma \vdash p \to \varphi}{\Gamma, p \vdash \varphi} \qquad \textbf{(W 0)} \quad \frac{\Gamma \vdash \mathbf{0}}{\Gamma, p \vdash \mathbf{0}}$$

are admissible.

PROOF. For **(ER →)**, we have $\varphi \vdash \varphi$ by Lemma 4.2. The following figure gives the remainder of the derivation.

$$\cfrac{\Gamma \vdash p \to \varphi \qquad \cfrac{\cfrac{\varphi \vdash \varphi}{p, \varphi \vdash \varphi}\ \textbf{(W}\ p\textbf{)}}{\vdots\ \textbf{(W}\ p\textbf{)}, \textbf{(W}\ \psi\textbf{)}}}{\cfrac{\Gamma, p, \varphi \vdash \varphi}{\Gamma, p \to \varphi, p \vdash \varphi}\ \textbf{(I}\to\textbf{)}}\ \textbf{(cut)}}{\Gamma, p \vdash \varphi}$$

To derive **(W 0)**, the sequent $\Gamma, \mathbf{0}, p \vdash \mathbf{0}$ is an instance of **(I 0)**. Applying **(cut)** to this and the premise $\Gamma \vdash \mathbf{0}$ yields the desired conclusion. □

The rule **(ER →)** plays the role of *Modus Ponens*. It is clearly an elimination rule on the right. We cannot write a real Modus Ponens rule in the present system, since a programs cannot stay to the right of $\vdash$. If this were allowed, **(ER →)** and Modus Ponens would have been easily interderivable. See also the proof of Lemma 4.4.

We wish to pause and discuss briefly why we view partial correctness reasoning in S as intuitionistic rather than classical. It is not immediately obvious, since formulas are of the form $p_1 \to \cdots \to p_n \to b$, where $p_1, \ldots, p_n$ are programs and b is a test. In particular, formulas are not closed under implication. But we can argue that the implication in the formula $p \to \varphi$ has an intuitionistic flavor by considering the rules that introduce implication. Rule **(R →)** is a typical rule of introduction of implication on the right of $\vdash$. Rule **(I →)** is not so typical, but it can be shown that this rule is derivable from **(ident)**, **(ER →)**, **(W ψ)**, **(W p)**, and **(cut)** as follows.

$$\dfrac{\dfrac{\dfrac{p \to \psi \vdash p \to \psi}{p \to \psi, p \vdash \psi}\ \textbf{(ER}\to\textbf{)}}{\vdots\ \textbf{(W}\ \psi\textbf{), (W}\ p\textbf{)}}}{\Gamma, p \to \psi, p \vdash \psi} \qquad \dfrac{\Gamma, p, \psi, \Delta \vdash \varphi}{\Gamma, p \to \psi, p, \psi, \Delta \vdash \varphi}\ \textbf{(W}\ \psi\textbf{)}$$
$$\dfrac{\qquad\qquad}{\Gamma, p \to \psi, p, \Delta \vdash \varphi}\ \textbf{(cut)}$$

Since each of the rules used in the above derivation clearly has an intuitionistic flavor, it follows that **(I →)** has as well.

Next we show that S is powerful enough to prove all classically valid tests.

LEMMA 4.4. *For tests b, c, the sequent $b \vdash c$ is derivable in* S *whenever $b \to c$ is a classical propositional tautology.*

PROOF. It is well known (see [Harel et al. 2000; Johnstone 1987]) that the following proof system is complete for classical propositional logic:

(MP) $\dfrac{\vdash b \quad \vdash b \to c}{\vdash c}$

(K) $\vdash b \to c \to b$

(S) $\vdash (b \to c \to d) \to (b \to c) \to (b \to d)$

(DN) $\vdash \overline{\overline{b}} \to b$

We show that **(MP)**–**(DN)** are derivable in S. For **(MP)**,

$$\dfrac{\vdash b \qquad \dfrac{\vdash b \to c}{b \vdash c}\ \textbf{(ER}\to\textbf{)}}{\vdash c}\ \textbf{(cut)}$$

For **(K)**,

$$\dfrac{\dfrac{b \vdash b}{b, c \vdash b}\ \textbf{(W}\ \psi\textbf{)}}{\vdash b \to c \to b}\ \textbf{(R}\to\textbf{), (R}\to\textbf{)}$$

For **(S)**,

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{d \vdash d}{c, d \vdash d}\,(\mathbf{W}\,\psi)}{c \to d, c \vdash d}\,(\mathbf{I}\to)}{b, c \to d, c \vdash d}\,(\mathbf{W}\,\psi)}{b \to c \to d, b, c \vdash d}\,(\mathbf{I}\to)}{b \to c \to d, b \to c, b \vdash d}\,(\mathbf{I}\to)}{\vdash (b \to c \to d) \to (b \to c) \to (b \to d)}\,(\mathbf{R}\to), (\mathbf{R}\to), (\mathbf{R}\to)$$

Finally, for **(DN)**,

$$\dfrac{\dfrac{\dfrac{\overline{b}, \mathbf{0} \vdash b}{\overline{b} \to \mathbf{0}, \overline{b} \vdash b}\,(\mathbf{I}\to) \quad \dfrac{b \vdash b}{\overline{\overline{b}}, b \vdash b}\,(\mathbf{W}\,\psi)}{\overline{\overline{b}} \vdash b}\,\textbf{(test-cut)}}{\vdash \overline{\overline{b}} \to b}\,(\mathbf{R}\to)$$

This completes the proof. □

LEMMA 4.5. *The rule*

$$\textbf{(iter)} \quad \frac{\varphi, p \vdash \varphi}{\varphi, p^+ \vdash \varphi}$$

is admissible.

PROOF. Immediate from $(\mathbf{I}^+)$ by taking $\psi = \varphi$. □

LEMMA 4.6. *The rules*

$$\textbf{(curry)} \quad \frac{\Gamma, p \to q \to \psi, \Delta \vdash \varphi}{\Gamma, pq \to \psi, \Delta \vdash \varphi}$$

$$\textbf{(uncurry)} \quad \frac{\Gamma, pq \to \psi, \Delta \vdash \varphi}{\Gamma, p \to q \to \psi, \Delta \vdash \varphi}$$

are admissible.

PROOF. By straightforward derivations involving **(cut)**, it suffices to show that both $pq \to \psi \vdash p \to q \to \psi$ and $p \to q \to \psi \vdash pq \to \psi$. For the former, starting with $pq \to \psi \vdash pq \to \psi$, apply **(ER →)** and **(E ⊗)** to get $pq \to \psi, p, q \vdash \psi$, then apply **(R →)** twice. For the latter, starting with $\psi \vdash \psi$, apply **(W *p*)** twice to get $p, q, \psi \vdash \psi$, then apply **(I →)** twice to get $p \to q \to \psi, p, q \vdash \psi$. The result then follows from **(I ⊗)** and **(R →)**. □

LEMMA 4.7. *Every φ is provably equivalent to some $p \to \mathbf{0}$ in the sense that $\varphi \vdash p \to \mathbf{0}$ and $p \to \mathbf{0} \vdash \varphi$.*

PROOF. The formula $q_1 \to \cdots \to q_n \to b$ is equivalent to $q_1 \cdots q_n \overline{b} \to \mathbf{0}$. The proof of this fact is quite easy using Lemma 4.6 and is left to the reader. □

4.2 Relation to Kleene Algebra

We show in this section that S induces a left-handed Kleene algebra structure on programs. The main result of this section, Proposition 4.13, relates provability in

System S with containment of regular sets of guarded strings. It plays the key role in the completeness proof of System S (Theorem 6.1).

Recall that a *Kleene algebra* (KA) is an idempotent semiring such that p^*q is the least solution to $q+px \le x$ and qp^* is the least solution to $q+xp \le x$. Equivalently, a Kleene algebra is a structure $(K,\ +,\ \cdot,\ ^*,\ 0,\ 1)$ satisfying the following axioms.

$$\begin{aligned}
p + (q + r) &= (p + q) + r \\
p + q &= q + p \\
p + 0 = p + p &= p \\
p(qr) &= (pq)r \\
1p = p1 &= p \\
p(q + r) &= pq + pr \\
(p + q)r &= pr + qr \\
0p = p0 &= 0 \\
1 + pp^* &= 1 + p^*p = p \qquad (1) \\
px \le x &\rightarrow p^*x \le x \qquad (2) \\
xp \le x &\rightarrow xp^* \le x. \qquad (3)
\end{aligned}$$

Boffa [1990; 1995], based on results of Krob [1991], shows that for the equational theory of the regular sets, the right-hand rule (3) is unnecessary. We will call an idempotent semiring satisfying (1) and (2) a *left-handed Kleene algebra*. Boffa's result says that for regular expressions p and q, $R(p) = R(q)$ iff $p = q$ is a logical consequence of the axioms of left-handed Kleene algebra, where R is the usual interpretation of regular expressions as sets of strings.

More specifically, Krob [1991] shows that the *classical equations* of Conway [1971], along with a certain infinite but independently characterized set of axioms, logically entail all identities of the regular sets over P. The classical equations of Conway are the axioms of idempotent semirings, the equations (1), and the equations

$$\begin{aligned}
(p + q)^* &= (p^*q)^*p^* \\
p^* &= p^{**} \\
(pq)^* &= 1 + p(qp)^*q \\
p^* &= (p^n)^*(1 + p)^{n-1},\ n \ge 0.
\end{aligned}$$

Boffa [1990; 1995] actually shows that these equations plus the rule

$$p^2 = p \rightarrow p^* = 1 + p \qquad (4)$$

—which, the reader will note, is neither left- nor right-handed—imply all the axioms of Krob, therefore the classical equations of Conway plus Boffa's rule (4) are complete for the equational theory of the regular sets over P. The classical equations and Boffa's rule are all easily shown to be theorems of left-handed KA.

Our first task is to extend these results to Kleene algebra with tests (KAT) and guarded strings. First let us recall that Kleene algebra with tests is a Kleene algebra with an embedded Boolean subalgebra. Formally, it is a two-sorted algebra

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

such that

—$(K, +, \cdot, {}^*, 0, 1)$ is a Kleene algebra

—$(B, +, \cdot, \bar{}, 0, 1)$ is a Boolean algebra

—$(B, +, \cdot, 0, 1)$ is a subalgebra of $(K, +, \cdot, 0, 1)$.

LEMMA 4.8. *Left-handed* KAT *is complete for the equational theory of the regular sets of guarded strings over* P *and* B*. In other words, for every pair of programs p, q in the language of* KAT*, $GS(p) = GS(q)$ if and only if the equation $p = q$ is a logical consequence of the axioms of left-handed* KAT*.*

PROOF. We adapt an argument of [Kozen and Smith 1996], in which the same result was proved for KAT with both the left- and right-hand rule. It was shown there that for any program p, there is an equivalent program $\widehat{p}$ such that

(i) $p = \widehat{p}$ is a theorem of KAT, and

(ii) $GS(\widehat{p}) = R(\widehat{p})$, where $R(\widehat{p})$ is the regular set of strings over the alphabet $\mathsf{P} \cup \mathsf{B} \cup \{\overline{b} \mid b \in \mathsf{B}\}$ denoted by $\widehat{p}$ under the usual interpretation of regular expressions.

In other words, any p can be transformed by the axioms of KAT to another program $\widehat{p}$ such that the set of guarded strings denoted by $\widehat{p}$ is the same as the set of strings denoted by $\widehat{p}$.

Now to show completeness of KAT over guarded strings, [Kozen and Smith 1996] argued as follows. Suppose $GS(p) = GS(q)$. Then

$$R(\widehat{p}) = GS(\widehat{p}) = GS(p) = GS(q) = GS(\widehat{q}) = R(\widehat{q}).$$

Since KA is complete for the equational theory of the regular sets, $\widehat{p} = \widehat{q}$ is a theorem of KA. Combining this with (i) for p and q implies that $p = q$ is a theorem of KAT.

To adapt this to the present situation, we observe that $\widehat{p} = \widehat{q}$ is a theorem of left-handed KA by the results of Boffa and Krob. Thus in order to complete the proof, we need only ascertain that the right-hand rule (3) is not needed in the proof of $p = \widehat{p}$. This does not follow from Boffa's and Krob's results, since the argument is in KAT, not KA. However, a perusal of [Kozen and Smith 1996] reveals that the proof of $p = \widehat{p}$ uses neither the left- nor the right-hand rule, but can be carried out using only the classical equations of Conway and the axioms of Boolean algebra. □

We now describe the left-handed KAT structure induced by S. For programs p, q we define $p \sqsubseteq q$ if $q \to \varphi \vdash p \to \varphi$ is admissible; that is, if $q \to \varphi \vdash p \to \varphi$ is provable for all φ. Define $p \equiv q$ if $p \sqsubseteq q$ and $q \sqsubseteq p$. The relation $\sqsubseteq$ is a preorder, therefore $\equiv$ is an equivalence relation and $\sqsubseteq$ is a partial order on $\equiv$-classes. Reflexivity is **(ident)** (Lemma 4.2) and transitivity follows from a single application of **(cut)**. The relation $\sqsubseteq$ is a proof-theoretic approximation of the relation of containment of the input-output relations denoted by programs p, q. Indeed, if in all models the meaning of p is always contained in the meaning of q, then for every post-condition φ, if p is partially correct with respect to φ, then so is q. In other words, the sequent $q \to \varphi \vdash p \to \varphi$ is valid. It turns out that this approximation is strong enough to induce a left-handed KAT structure on $\equiv$-classes of programs.

LEMMA 4.9. *The operators $\oplus$ and $\otimes$ are monotone with respect to $\sqsubseteq$. That is, if $p \sqsubseteq q$, then $p \oplus r \sqsubseteq q \oplus r$, $pr \sqsubseteq qr$, and $rp \sqsubseteq rq$.*

PROOF. The rules **(E1 $\oplus$)**, **(E2 $\oplus$)**, and **(I $\oplus$)** imply that $p \oplus q$ is the $\sqsubseteq$-least upper bound of p and q modulo $\equiv$. The monotonicity of $\oplus$ follows by equational reasoning:

$$p \sqsubseteq q \;\Rightarrow\; p \sqsubseteq q \oplus r \text{ and } r \sqsubseteq q \oplus r \;\Rightarrow\; p \oplus r \sqsubseteq q \oplus r.$$

For $\otimes$, we must show that if $q \to \varphi \vdash p \to \varphi$ for any φ, then $qr \to \varphi \vdash pr \to \varphi$ and $rq \to \varphi \vdash rp \to \varphi$ for any φ. Using **(cut)**, **(curry)**, and **(uncurry)** (Lemma 4.6), it suffices to show that $q \to r \to \varphi \vdash p \to r \to \varphi$ and $r \to q \to \varphi \vdash r \to p \to \varphi$ for any φ. The former is immediate from the assumption, and the latter follows from **(mono)** (Lemma 4.2). □

LEMMA 4.10. *If $p \sqsubseteq q$ and $qq \sqsubseteq q$, then $p^+ \sqsubseteq q$.*

PROOF. Certainly $pq \sqsubseteq q$ by monotonicity. Then

$$\dfrac{\dfrac{\dfrac{q \to \varphi \vdash p \to \varphi}{q \to \varphi, p \vdash \varphi}\ \textbf{(ER}\to\textbf{)} \qquad \dfrac{\dfrac{q \to \varphi \vdash pq \to \varphi}{q \to \varphi, pq \vdash \varphi}\ \textbf{(E}\otimes\textbf{)}}{q \to \varphi, p, q \vdash \varphi}\ \textbf{(R}\to\textbf{)}}{q \to \varphi, p \vdash q \to \varphi}\ \textbf{(I}^+\textbf{)}}{q \to \varphi, p^+ \vdash \varphi}\ \textbf{(R}\to\textbf{)}$$
$$q \to \varphi \vdash p^+ \to \varphi$$

□

LEMMA 4.11. *Let $\mathcal{P}/\equiv$ denote the set of $\equiv$-equivalence classes. The operations $\oplus$, $\otimes$, and * are well defined on $\mathcal{P}/\equiv$, and the quotient structure $(\mathcal{P}/\equiv, \oplus, \otimes, ^*, \mathbf{0}, \mathbf{1})$ is a left-handed* KA.

PROOF. We must argue that all the following properties hold:

$p \oplus (q \oplus r) \equiv (p \oplus q) \oplus r$	$p(qr) \equiv (pq)r$
$p \oplus q \equiv q \oplus p$	$\mathbf{1}p \equiv p\mathbf{1} \equiv p$
$p \oplus \mathbf{0} \equiv p$	$\mathbf{0}p \equiv p\mathbf{0} \equiv \mathbf{0}$
$p \oplus p \equiv p$	$\mathbf{1} \oplus pp^* \equiv p^*$
$p(q \oplus r) \equiv pq \oplus pr$	$\mathbf{1} \oplus p^*p \equiv p^*$
$(p \oplus q)r \equiv pr \oplus qr$	$pq \sqsubseteq q \Rightarrow p^*q \sqsubseteq q.$

These are just the laws of left-handed KA written with the symbols of S.

To derive the distributive law

$$p(q \oplus r) \;\sqsubseteq\; pq \oplus pr,$$

first from **(ER $\to$)**, **(E1 $\oplus$)**, and **(E $\otimes$)**, one can derive $pq \oplus pr \to \varphi, p, q \vdash \varphi$ from $pq \oplus pr \to \varphi \vdash pq \oplus pr \to \varphi$. Similarly, one can derive $pq \oplus pr \to \varphi, p, r \vdash \varphi$ using **(E2 $\oplus$)** instead of **(E1 $\oplus$)**. Then

$$\dfrac{\dfrac{pq \oplus pr \to \varphi, p, q \vdash \varphi \qquad pq \oplus pr \to \varphi, p, r \vdash \varphi}{pq \oplus pr \to \varphi, p, q \oplus r \vdash \varphi}\ \textbf{(I}\oplus\textbf{)}}{pq \oplus pr \to \varphi \vdash p(q \oplus r) \to \varphi}\ \textbf{(I}\otimes\textbf{), (R}\to\textbf{)}$$

All the other axioms of idempotent semirings follow in an equally straightforward manner. Since $\oplus$ and $\otimes$ are monotone with respect to $\sqsubseteq$ (Lemma 4.9), they are well defined on $\equiv$-classes.

The inequality $p^+p^+ \sqsubseteq p^+$ follows from $\mathbf{(CC^+)}$ by:

$$\dfrac{\dfrac{\dfrac{p^+ \to \varphi \vdash p^+ \to \varphi}{p^+ \to \varphi, p^+ \vdash \varphi}\ \mathbf{(ER\to)}}{p^+ \to \varphi, p^+, p^+ \vdash \varphi}\ \mathbf{(CC^+)}}{p^+ \to \varphi \vdash p^+p^+ \to \varphi}\ \mathbf{(I\otimes),\ (R\to)}$$

The inequality $p \sqsubseteq p^+$ follows from $\mathbf{(E^+)}$ in a similar fashion. Monotonicity of $^+$ and * then follow from Lemma 4.10 by equational reasoning:

$$p \sqsubseteq q \ \Rightarrow\ p \sqsubseteq q^+ \text{ and } q^+q^+ \sqsubseteq q^+ \Rightarrow p^+ \sqsubseteq q^+$$

$$p \sqsubseteq q \ \Rightarrow\ p^* = \mathbf{1} \oplus p^+ \sqsubseteq \mathbf{1} \oplus q^+ = q^*.$$

We now prove the KA identities involving *. Arguing equationally, we have

$$p \oplus pp^+ \ \sqsubseteq\ p^+ \oplus p^+p^+ \ \sqsubseteq\ p^+ \oplus p^+ \ \sqsubseteq\ p^+,$$

and similarly $p \oplus p^+p \sqsubseteq p^+$. For the opposite inequalities we will use Lemma 4.10. Clearly we have $p \sqsubseteq p \oplus pp^+$. We also have $pp \sqsubseteq pp^+$, $ppp^+ \sqsubseteq pp^+$, $pp^+p \sqsubseteq pp^+$ and $pp^+pp^+ \sqsubseteq pp^+$, hence

$$(p \oplus pp^+)(p \oplus pp^+) \ \sqsubseteq\ pp^+ \ \sqsubseteq\ p \oplus pp^+.$$

By Lemma 4.10, $p^+ \sqsubseteq p \oplus pp^+$. Since the opposite inequality was already established, we have $p^+ \equiv p \oplus pp^+$.

Now we can show that $\mathbf{1} \oplus pp^* \equiv p^*$:

$$\begin{aligned} p^* \ &\equiv\ \mathbf{1} \oplus p^+ \ \equiv\ \mathbf{1} \oplus p \oplus pp^+ \ \equiv\ \mathbf{1} \oplus p(\mathbf{1} \oplus p^+) \\ &\equiv\ \mathbf{1} \oplus pp^*. \end{aligned}$$

The identities $p^+ \equiv p \oplus p^+p$ and $\mathbf{1} \oplus p^*p \equiv p^*$ are obtained in a similar fashion.

It remains to show $pq \sqsubseteq q \Rightarrow p^*q \sqsubseteq q$. This is established by the following derivation:

$$\dfrac{\dfrac{q \to \varphi \vdash q \to \varphi}{q \to \varphi, \mathbf{1} \vdash q \to \varphi}\ \mathbf{(W\,\psi)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{q \to \varphi \vdash pq \to \varphi}{q \to \varphi, pq \vdash \varphi}\ \mathbf{(ER\to)}}{q \to \varphi, p, q \vdash \varphi}\ \mathbf{(E\otimes)}}{q \to \varphi, p \vdash q \to \varphi}\ \mathbf{(R\to)}}{q \to \varphi, p^+ \vdash q \to \varphi}\ \mathbf{(iter)}}{\dfrac{q \to \varphi, \mathbf{1} \oplus p^+ \vdash q \to \varphi}{q \to \varphi \vdash (\mathbf{1} \oplus p^+)q \to \varphi}\ \mathbf{(I\oplus)}}\ \mathbf{(ER\to),\ (I\otimes),\ (R\to)}$$

□

LEMMA 4.12. *If $b \to c$ is a classical tautology, then $b \sqsubseteq c$. Thus the tests form a Boolean algebra modulo $\equiv$.*

PROOF. We have $c \to \varphi, b \vdash c$ by the axiom $b \vdash c$ and the weakening rule **(W ψ)**, and we have $c \to \varphi, c \vdash \varphi$ by **(ER $\to$)**. The desired conclusion $c \to \varphi \vdash b \to \varphi$ then follows from **(cut)** and **(R $\to$)**. □

Combining Lemmas 4.11 and 4.12 and the fact that the regular sets of guarded strings form the free KAT on generators P and B, we have

PROPOSITION 4.13. *The structure* $(\mathcal{P}/\equiv, \mathcal{B}/\equiv, \oplus, \otimes, {}^*, \overline{\ }, \mathbf{0}, \mathbf{1})$ *is a left-handed* KAT *and is isomorphic to the algebra of regular sets of guarded strings over* P *and* B. *Thus for any programs p and q, $p \sqsubseteq q$ iff $GS(p) \subseteq GS(q)$ and $p \equiv q$ iff $GS(p) = GS(q)$.*

PROOF. In order to show that $(\mathcal{P}/\equiv, \mathcal{B}/\equiv, \oplus, \otimes, {}^*, \overline{\ }, \mathbf{0}, \mathbf{1})$ is a left-handed KAT, by Lemmas 4.11 and 4.12, it remains to show that $\otimes$ coincides in $\mathcal{B}/\equiv$ with the greatest lower bound and that $\oplus$ coincides with the least upper bound. The greatest lower bound in $\mathcal{B}/\equiv$ of two equivalence classes with representatives b and c is the equivalence class of $\overline{b \to \overline{c}}$, while their least upper bound is the equivalence class of $\overline{b} \to c$. Hence we have to prove the following two equivalences.

$$b \oplus c \equiv \overline{b} \to c \tag{5}$$

$$b \otimes c \equiv \overline{(b \to \overline{c})} \tag{6}$$

We start with (5). Since $b \to \overline{b} \to c$ is a propositional tautology, it follows from Lemma 4.4 that $b \vdash \overline{b} \to c$ is derivable. Hence

$$\dfrac{\dfrac{b \vdash \overline{b} \to c}{(\overline{b} \to c) \to \varphi, b \vdash \overline{b} \to c}\ \textbf{(W }\psi\textbf{)} \qquad \dfrac{\dfrac{(\overline{b} \to c) \to \varphi \vdash (\overline{b} \to c) \to \varphi}{(\overline{b} \to c) \to \varphi, \overline{b} \to c \vdash \varphi}\ \textbf{(ER }\to\textbf{)}}{(\overline{b} \to c) \to \varphi, b, \overline{b} \to c \vdash \varphi}\ \textbf{(W }\psi\textbf{)}}{(\overline{b} \to c) \to \varphi, b \vdash \varphi}\ \textbf{(cut)}$$

In a similar way, since $c \to \overline{b} \to c$ is a propositional tautology, we derive the sequent $(\overline{b} \to c) \to \varphi, c \vdash \varphi$. Hence, by **(I $\oplus$)** and **(R $\to$)**, we obtain

$$(\overline{b} \to c) \to \varphi \vdash b \oplus c \to \varphi$$

i.e. $b \oplus c \sqsubseteq \overline{b} \to c$.

The opposite inequality is established by the following derivation.

$$\dfrac{\dfrac{\dfrac{\dfrac{b \oplus c \to \varphi \vdash b \oplus c \to \varphi}{b \oplus c \to \varphi, b \oplus c \vdash \varphi}\ \textbf{(ER }\to\textbf{)}}{b \oplus c \to \varphi, b \vdash \varphi}\ \textbf{(E1 }\oplus\textbf{)}}{b \oplus c \to \varphi, \overline{b} \to c, b \vdash \varphi}\ \textbf{(I }\to\textbf{)} \qquad \dfrac{\dfrac{\dfrac{b \oplus c \to \varphi \vdash b \oplus c \to \varphi}{b \oplus c \to \varphi, b \oplus c \vdash \varphi}\ \textbf{(E2 }\oplus\textbf{)}}{b \oplus c \to \varphi, c \vdash \varphi}\ \textbf{(W }\psi\textbf{)}}{b \oplus c \to \varphi, \overline{b}, c \vdash \varphi}\ \textbf{(I }\to\textbf{)}}{b \oplus c \to \varphi, \overline{b} \to c, \overline{b} \vdash \varphi}\ \textbf{(test-cut)}$$

$$\dfrac{b \oplus c \to \varphi, \overline{b} \to c \vdash \varphi}{b \oplus c \to \varphi \vdash (\overline{b} \to c) \to \varphi}\ \textbf{(R }\to\textbf{)}$$

This proves (5).

For the proof of $\sqsubseteq$ in (6) let us observe that $b \to c \to \overline{(b \to \overline{c})}$ is a propositional tautology. Hence by Lemma 4.4 and **(ER $\to$)** we obtain

$$b, c \vdash \overline{(b \to \overline{c})}.$$

The rest of the derivation follows.

$$\dfrac{\dfrac{\dfrac{b, c \vdash \overline{(b \to \overline{c})}}{\overline{(b \to \overline{c})} \to \varphi, b, c \vdash \overline{(b \to \overline{c})}}\ (\mathbf{W}\,\psi) \quad \dfrac{\dfrac{\overline{(b \to \overline{c})} \to \varphi \vdash \overline{(b \to \overline{c})} \to \varphi}{\overline{(b \to \overline{c})} \to \varphi, \overline{(b \to \overline{c})} \vdash \varphi}\ (\mathbf{ER}\to)}{\overline{(b \to \overline{c})} \to \varphi, b, c, \overline{(b \to \overline{c})} \vdash \varphi}\ (\mathbf{W}\,\psi),\ (\mathbf{W}\,\psi)}{\dfrac{\overline{(b \to \overline{c})} \to \varphi, b, c \vdash \varphi}{\overline{(b \to \overline{c})} \to \varphi, b \otimes c \vdash \varphi}\ (\mathbf{I}\otimes)}\ (\mathbf{cut})$$
$$\dfrac{\overline{(b \to \overline{c})} \to \varphi, b \otimes c \vdash \varphi}{\overline{(b \to \overline{c})} \to \varphi \vdash b \otimes c \to \varphi}\ (\mathbf{R}\to)$$

For the opposite inequality, let us observe that $\overline{(b \to \overline{c})} \to b$ and $\overline{(b \to \overline{c})} \to c$ are propositional tautologies. Hence by Lemma 4.4 and $(\mathbf{W}\,\psi)$, we can assume $b \otimes c \to \varphi, \overline{(b \to \overline{c})} \vdash b$ and $b \otimes c \to \varphi, \overline{(b \to \overline{c})} \vdash c$. The rest of the derivation follows.

$$\dfrac{\dfrac{b \otimes c \to \varphi, \overline{(b \to \overline{c})} \vdash c \qquad \dfrac{b \otimes c \to \varphi, \overline{(b \to \overline{c})} \vdash b \quad \dfrac{\dfrac{\dfrac{b \otimes c \to \varphi \vdash b \otimes c \to \varphi}{b \otimes c \to \varphi, b \otimes c \vdash \varphi}\ (\mathbf{ER}\to)}{b \otimes c \to \varphi, b, c \vdash \varphi}\ (\mathbf{E}\otimes)}{b \otimes c \to \varphi, \overline{(b \to \overline{c})}, b, c \vdash \varphi}\ (\mathbf{W}\,\psi)}{b \otimes c \to \varphi, \overline{(b \to \overline{c})}, c \vdash \varphi}\ (\mathbf{cut})}{b \otimes c \to \varphi, \overline{(b \to \overline{c})} \vdash \varphi}\ (\mathbf{cut})$$
$$\dfrac{b \otimes c \to \varphi, \overline{(b \to \overline{c})} \vdash \varphi}{b \otimes c \to \varphi \vdash \overline{(b \to \overline{c})} \to \varphi}\ (\mathbf{R}\to)$$

It remains to argue that the quotient structure $(\mathcal{P}/\equiv,\ \mathcal{B}/\equiv)$ and the algebra of regular sets of guarded strings over P and B are isomorphic. By Lemma 4.8, KAT and left-handed KAT have the same equational theory, thus the guarded string algebra, being the free KAT on generators P, B [Kozen and Smith 1996], is also the free left-handed KAT on generators P, B. Since the structure $(\mathcal{P}/\equiv,\ \mathcal{B}/\equiv)$ is a left-handed KAT, it satisfies all the equations of left-handed KAT under its canonical interpretation, thus $GS(p) = GS(q)$ implies $p \equiv q$.

Conversely, suppose $p \sqsubseteq q$. Then $q \to \varphi \vdash p \to \varphi$ for all formulas φ; in particular, $q \to b \vdash p \to b$ for atomic b not occurring in p or q. By the soundness of System S (Theorem 5.1 below), this sequent is valid in all relation algebras. In the notation of Dynamic Logic [Harel et al. 2000], this sequent is expressed $[([q]b)?]\,[p]b$, which is equivalent to $[q]b \to [p]b$. By [Harel et al. 2000, Exercise 5.3, p. 188], $[p]_K \subseteq [q]_K$ in all relational models of all Kripke frames K. Since the guarded string model is isomorphic to a relational model [Kozen and Smith 1996, Lemma 5], $GS(p) \subseteq GS(q)$. □

4.3 Incompleteness of Hoare Logic

The partial correctness assertion $\{b\}\,p\,\{c\}$ of HL is encoded in S by the formula $b \to p \to c$. The Hoare-style rule

$$\frac{\{b_1\}\,p_1\,\{c_1\},\ \ldots,\ \{b_n\}\,p_n\,\{c_n\}}{\{b\}\,p\,\{c\}} \tag{7}$$

is encoded by the sequent

$$b_1 \to p_1 \to c_1, \ldots, b_n \to p_n \to c_n \ \vdash\ b \to p \to c.$$

It follows from Theorem 6.1 that all relationally valid rules of this form are derivable in S; this is false for HL (see [Kozen 2000; Kozen and Tiuryn 2001]). We give two examples of this situation. First let us remark that every relationally valid partial correctness assertion $\{b\}\,p\,\{c\}$ is derivable in Hoare logic. Recall that we are dealing with a propositional formalism, hence the incompleteness arguments for first-order Hoare logic do not apply here. Derivability in Hoare logic of relationally valid partial correctness assertions follows from a more general result. It is shown in [Kozen and Tiuryn 2001, Theorem 4.1] that every relationally valid rule (7) in which the programs $p_1, \ldots, p_n$ are atomic is derivable in Hoare logic. Since a single partial correctness assertion is a special case of rule (7) for $n = 0$, the above remark follows. Thus we have to look for examples of relationally valid rules (7) with non-atomic premises. One such rule, mentioned in [Kozen and Tiuryn 2001], that is relationally valid but not derivable in Hoare logic is

$$\frac{\{b\}\,p^*\,\{c\}}{\{b\}\,p\,\{c\}}. \tag{8}$$

The sequent of S corresponding to (8) is $b \to (\mathbf{1} \oplus p^+) \to c \vdash b \to p \to c$. Here is a derivation of this sequent:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\mathbf{1} \oplus p^+) \to c \vdash (\mathbf{1} \oplus p^+) \to c}{(\mathbf{1} \oplus p^+) \to c, \mathbf{1} \oplus p^+ \vdash c}\ \textbf{(ER}\to\textbf{)}}{(\mathbf{1} \oplus p^+) \to c, p^+ \vdash c}\ \textbf{(E2}\oplus\textbf{)}}{(\mathbf{1} \oplus p^+) \to c, p \vdash c}\ \textbf{(E}^{+}\textbf{)}}{b, (\mathbf{1} \oplus p^+) \to c, p \vdash c}\ \textbf{(W}\,\psi\textbf{)}}{b \to (\mathbf{1} \oplus p^+) \to c, b, p \vdash c}\ \textbf{(I}\to\textbf{)}}{b \to (\mathbf{1} \oplus p^+) \to c \vdash b \to p \to c}\ \textbf{(R}\to\textbf{), (R}\to\textbf{)}$$

The first sequent in the above derivation is an instance of **(ident)** (Lemma 4.2).

Another example of a relationally valid rule which is not derivable in HL is

$$\frac{\{d\}\ \textbf{if}\ b\ \textbf{then}\ p\ \textbf{else}\ p\,\{c\}}{\{d\}\,p\,\{c\}}. \tag{9}$$

The reason that the above rule cannot be derived is the same as for (8): it is easy to show by induction on proofs in Hoare logic that no conclusion with an atomic program can be derived from non-atomic premises. The program **if** b **then** p **else** p is encoded in S by $bp \oplus \overline{b}p$. Here is a derivation in S of the sequent corresponding to the rule (9):

$$\dfrac{\dfrac{\dfrac{\dfrac{(bp \oplus \overline{b}p) \to c \vdash (bp \oplus \overline{b}p) \to c}{(bp \oplus \overline{b}p) \to c, bp \oplus \overline{b}p \vdash c}\ \textbf{(ER}\to\textbf{)}}{(bp \oplus \overline{b}p) \to c, bp \vdash c}\ \textbf{(E1}\oplus\textbf{)}}{(bp \oplus \overline{b}p) \to c, b, p \vdash c}\ \textbf{(E}\otimes\textbf{)} \qquad \dfrac{\dfrac{\dfrac{(bp \oplus \overline{b}p) \to c \vdash (bp \oplus \overline{b}p) \to c}{(bp \oplus \overline{b}p) \to c, bp \oplus \overline{b}p \vdash c}\ \textbf{(ER}\to\textbf{)}}{(bp \oplus \overline{b}p) \to c, \overline{b}p \vdash c}\ \textbf{(E2}\oplus\textbf{)}}{(bp \oplus \overline{b}p) \to c, \overline{b}, p \vdash c}\ \textbf{(E}\otimes\textbf{)}}{\dfrac{(bp \oplus \overline{b}p) \to c, p \vdash c}{(bp \oplus \overline{b}p) \to c \vdash p \to c}\ \textbf{(R}\to\textbf{)}}\ \textbf{(test-cut)}$$

$$\frac{(bp \oplus \overline{b}p) \to c \vdash p \to c}{d \to (bp \oplus \overline{b}p) \to c \vdash d \to p \to c}\ \textbf{(mono)}$$

5. SOUNDNESS

THEOREM 5.1. *If $\Gamma \vdash \varphi$ is provable, then it is valid in all trace and relational models.*

PROOF. By Corollary 3.4, we need only to show soundness over trace models. This is easily established by induction on proofs in S with one case for each proof rule. We argue the cases **(cut)** and $(\mathbf{I} \rightarrow)$ explicitly.

For **(cut)**, we need to show that

$$[\![\Gamma, \Delta]\!]_K \subseteq [\![\Gamma, \Delta, \varphi]\!]_K$$

under the assumptions

$$\begin{aligned} [\![\Gamma]\!]_K &\subseteq [\![\Gamma, \psi]\!]_K \\ [\![\Gamma, \psi, \Delta]\!]_K &\subseteq [\![\Gamma, \psi, \Delta, \varphi]\!]_K. \end{aligned}$$

Using monotonicity of $\circ$,

$$\begin{aligned} & [\![\Gamma, \Delta]\!]_K \\ &= [\![\Gamma]\!]_K \circ [\![\Delta]\!]_K \\ &\subseteq [\![\Gamma, \psi]\!]_K \circ [\![\Delta]\!]_K \\ &= [\![\Gamma, \psi, \Delta]\!]_K \\ &\subseteq [\![\Gamma, \psi, \Delta, \varphi]\!]_K \\ &= [\![\Gamma]\!]_K \circ [\![\psi]\!]_K \circ [\![\Delta, \varphi]\!]_K \\ &\subseteq [\![\Gamma]\!]_K \circ [\![\mathbf{1}]\!]_K \circ [\![\Delta, \varphi]\!]_K \\ &= [\![\Gamma]\!]_K \circ [\![\Delta, \varphi]\!]_K \\ &= [\![\Gamma, \Delta, \varphi]\!]_K. \end{aligned}$$

For $(\mathbf{I} \rightarrow)$, we want to show that if

$$[\![\Gamma, p, \psi, \Delta]\!]_K \subseteq \mathbf{last}^{-1}([\![\varphi]\!]_K),$$

then

$$[\![\Gamma, p \rightarrow \psi, p, \Delta]\!]_K \subseteq \mathbf{last}^{-1}([\![\varphi]\!]_K).$$

It suffices to show that

$$[\![p \rightarrow \psi]\!]_K \circ [\![p]\!]_K \subseteq [\![p]\!]_K \circ [\![\psi]\!]_K.$$

But

$$\begin{aligned} & \tau \in [\![p \rightarrow \psi]\!]_K \circ [\![p]\!]_K \\ &\Leftrightarrow \mathbf{first}(\tau) \in [\![p \rightarrow \psi]\!]_K \text{ and } \tau \in [\![p]\!]_K \\ &\Rightarrow \tau \in [\![p]\!]_K \text{ and } \mathbf{last}(\tau) \in [\![\psi]\!]_K \\ &\Leftrightarrow \tau \in [\![p]\!]_K \circ [\![\psi]\!]_K. \end{aligned}$$

The other cases are equally straightforward. □

6. COMPLETENESS

THEOREM 6.1. *If $\Gamma \not\vdash \varphi$, then there exist an acyclic trace model K and a trace $\sigma \in [\![\Gamma]\!]_K$ such that $\mathbf{last}(\sigma) \notin [\![\varphi]\!]_K$.*

PROOF. By Lemma 4.7, we can assume without loss of generality that φ is of the form $p \to \mathbf{0}$. The proof proceeds by induction on the length of Γ. For the basis of the induction, suppose Γ is empty, so that $\not\vdash p \to \mathbf{0}$. Then $p \not\equiv \mathbf{0}$. By Proposition 4.13, $GS(p) \neq \varnothing$. Construct a Kripke frame K consisting of a single acyclic trace σ such that $\mathrm{gs}(\sigma) \in GS(p)$. By Lemma 3.2, $\sigma \in [\![p]\!]_K$. Then $\mathbf{first}(\sigma) \in [\![\varepsilon]\!]_K$ and $\mathbf{first}(\sigma) \notin [\![p \to \mathbf{0}]\!]_K$.

For the induction step in which the environment ends with a program, say $\Gamma, p \not\vdash \varphi$, we have $\Gamma \not\vdash p \to \varphi$ by **(ER →)**. Applying the induction hypothesis, there exist an acyclic trace model K and traces σ and τ such that $\sigma \in [\![\Gamma]\!]_K$, $\mathbf{last}(\sigma) = \mathbf{first}(\tau)$, $\tau \in [\![p]\!]_K$, and $\mathbf{last}(\tau) \notin [\![\varphi]\!]_K$. Then $\sigma\tau \in [\![\Gamma, p]\!]_K$ and $\mathbf{last}(\sigma\tau) \notin [\![\varphi]\!]_K$.

Finally, we argue the induction step in which the environment ends with a formula, say $\Gamma, \psi \not\vdash \varphi$. By Lemma 4.7, we can rewrite this as $\Gamma, q \to \mathbf{0} \not\vdash p \to \mathbf{0}$. Let w be an expression representing the set of all guarded strings (see Lemma 3.1). Let r and s be programs such that $GS(r) = GS(p) \cap GS(qw)$ and $GS(s) = GS(p) - GS(qw)$. These programs exist by Lemma 3.1, and $GS(p) = GS(r \oplus s)$. By Proposition 4.13, we can replace p by $r \oplus s$ to get $\Gamma, q \to \mathbf{0} \not\vdash r \oplus s \to \mathbf{0}$. By **(R →)**, $\Gamma, q \to \mathbf{0}, r \oplus s \not\vdash \mathbf{0}$, and by **(I ⊕)**, either $\Gamma, q \to \mathbf{0}, r \not\vdash \mathbf{0}$ or $\Gamma, q \to \mathbf{0}, s \not\vdash \mathbf{0}$. But it cannot be the former, since $\Gamma, q \to \mathbf{0}, q, w \vdash \mathbf{0}$, therefore $\Gamma, q \to \mathbf{0} \vdash qw \to \mathbf{0}$, and by Proposition 4.13, $r \sqsubseteq qw$, therefore by **(cut)**, $\Gamma, q \to \mathbf{0} \vdash r \to \mathbf{0}$.

Thus it must be the case that $\Gamma, q \to \mathbf{0}, s \not\vdash \mathbf{0}$, so $\Gamma, q \to \mathbf{0} \not\vdash s \to \mathbf{0}$. By weakening we have $\Gamma \not\vdash s \to \mathbf{0}$. Then by the induction hypothesis, there exist an acyclic trace model K and traces $\sigma \in [\![\Gamma]\!]_K$ and $\tau \in [\![s]\!]_K$ such that $\mathbf{last}(\sigma) = \mathbf{first}(\tau)$. Construct a trace model M consisting only of the acyclic trace $\sigma\tau$. By Lemma 3.2, $\tau \notin [\![qw]\!]_M$, therefore no prefix of τ is in $[\![q]\!]_M$. Then $\mathbf{last}(\sigma) \in [\![q \to \mathbf{0}]\!]_M$, therefore $\sigma \in [\![\Gamma, q \to \mathbf{0}]\!]_M$. Moreover, $\mathbf{last}(\sigma) \notin [\![p \to \mathbf{0}]\!]_M$, since $\mathbf{last}(\sigma) = \mathbf{first}(\tau)$ and $\tau \in [\![p]\!]_M$. □

7. CONCLUSIONS AND FUTURE WORK

It has recently been shown that deciding whether a given sequent is valid is *PSPACE*-complete [Kozen 2001]. Several interesting questions present themselves for further investigation.

(1) The completeness proof relies on the results of Boffa [1990; 1995], which are based in turn on the results of Krob [1991]. Krob's proof is fairly involved, comprising an entire journal issue. One would like to have a proof of completeness based on first principles.

(2) The relative expressive and deductive power of S compared with similar systems such as KAT, PDL, and PHL is not completely understood. S is at least as expressive as PHL and the equational theory of KAT, and apparently more so, since it is not clear how to express general sequents $\varphi_1, p_1, \varphi_2, \ldots, p_{n-1}, \varphi_n \vdash \psi$ in PHL or KAT. On the other hand, it is not clear how to express general Horn formulas of KA such as $px = xq \to p^*x = xq^*$ in S.

(3) Application of the linear implication operator $\rightarrow$ is limited to programs on the left-hand side and formulas on the right-hand side. It would be interesting to see whether more general forms correspond to anything useful and whether the system can be extended to handle them. The operator $\rightarrow$ is a form of residuation (see [Pratt 1990; Kozen 1994]), and this connection bears further investigation. Also the issue of cut elimination in such a more general system seems to be an interesting problem. A more general implication would possibly allow the formulation of a system more in the style of Gentzen than System S.

(4) We would like to extend S to handle liveness properties and total correctness.

(5) We would like to undertake a deeper investigation into the structure of proofs with an eye toward establishing normal form and cut elimination theorems. It can be shown that the sequent $(b \oplus \overline{b}) \rightarrow \varphi \vdash \varphi$ cannot be derived in S without both **(cut)** and **(test-cut)**. Presumably it cannot be derived without **(test-cut)**. We do not know whether there are valid sequents that are not derivable in S without **(cut)**.

ACKNOWLEDGMENT

We thank Riccardo Pucella for pointing out an error in an earlier draft and the anonymous reviewers for their valuable comments.

REFERENCES

Artemov, S. 2001. Explicit provability and constructive semantics. *Bull. Symbolic Logic 7,* 1 (March), 1–36.

Boffa, M. 1990. Une remarque sur les systèmes complets d'identités rationnelles. *Informatique Théoretique et Applications/Theoretical Informatics and Applications 24,* 4, 419–423.

Boffa, M. 1995. Une condition impliquant toutes les identités rationnelles. *Informatique Théoretique et Applications/Theoretical Informatics and Applications 29,* 6, 515–518.

Conway, J. H. 1971. *Regular Algebra and Finite Machines.* Chapman and Hall, London.

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

Girard, J.-Y. 1987. Linear logic. *Theoretical Computer Science 50*, 1–102.

Gödel, K. 1933. Eine Interpretation des intuitionistischen Aussagenkalküls. *Ergebnisse eines mathematischen Kolloquiums 4*, 39–40. Reprinted in: S. Feferman, ed., Collected Works of Kurt Gödel, v. 1, New York, Oxford University Press, 1986.

Harel, D., Kozen, D., and Tiuryn, J. 2000. *Dynamic Logic.* MIT Press, Cambridge, MA.

Johnstone, P. 1987. *Notes on Logic and Set Theory.* Cambridge Mathematical Textbooks. Cambridge University Press.

Kaplan, D. M. 1969. Regular expressions and the equivalence of programs. *J. Comput. Syst. Sci. 3*, 361–386.

Kozen, D. 1994. On action algebras. In *Logic and Information Flow*, J. van Eijck and A. Visser, Eds. MIT Press, 78–88.

Kozen, D. 1997. Kleene algebra with tests. *Transactions on Programming Languages and Systems 19,* 3 (May), 427–443.

Kozen, D. 2000. On Hoare logic and Kleene algebra with tests. *Trans. Computational Logic 1,* 1 (July), 60–76.

Kozen, D. 2001. Automata on guarded strings and applications. Tech. Rep. 2001-1833, Computer Science Department, Cornell University. January.

Kozen, D. and Patron, M.-C. 2000. Certification of compiler optimizations using Kleene algebra with tests. In *Proc. 1st Int. Conf. Computational Logic (CL2000)* (London), J. Lloyd, V. Dahl,

U. Furbach, M. Kerber, K.-K. Lau, C. Palamidessi, L. M. Pereira, Y. Sagiv, and P. J. Stuckey, Eds. Lecture Notes in Artificial Intelligence, vol. 1861. Springer-Verlag, London, 568–582.

Kozen, D. and Smith, F. 1996. Kleene algebra with tests: Completeness and decidability. In *Proc. 10th Int. Workshop Computer Science Logic (CSL'96)*, D. van Dalen and M. Bezem, Eds. Lecture Notes in Computer Science, vol. 1258. Springer-Verlag, Utrecht, The Netherlands, 244–259.

Kozen, D. and Tiuryn, J. 2001. On the completeness of propositional Hoare logic. *Information Sciences 139,* 3–4, 187–195.

Kripke, S. 1963. Semantic analysis of modal logic. *Zeitschr. f. math. Logik und Grundlagen d. Math. 9*, 67–96.

Kripke, S. 1965. Semantical analysis of intuitionistic logic I. In *Formal Systems and Recursive Functions*, J. N. Crossley and M. A. E. Dummett, Eds. North-Holland, 92–130.

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

Pratt, V. 1990. Action logic and pure induction. In *Proc. Logics in AI: European Workshop JELIA '90*, J. van Eijck, Ed. Lecture Notes in Computer Science, vol. 478. Springer-Verlag, New York, 97–120.

Restall, G. 2000. *An Introduction to Substructural Logics*. Routledge.

Troelstra, A. S. 1992. *Lectures on Linear Logic*. CSLI Lecture Notes, vol. 29. Center for the Study of Language and Information.

Yetter, D. N. 1990. Quantales and (noncommutative) linear logic. *J. Symbolic Logic 55*, 41–64.

Received September 2001; revised June 2002; accepted June 2002