Introduction to Kleene Algebra **Lecture 14**
CS786 Spring 2004 **March 15, 2004**

KAT and Hoare Logic

In this lecture and the next we show that KAT subsumes propositional Hoare logic (PHL). Thus the specialized syntax and deductive apparatus of Hoare logic are inessential and can be replaced by simple equational reasoning. Moreover, all relationally valid Hoare-style inference rules are derivable in KAT (this is false for PHL). The results of this lecture are from [8, 7]. In a future lecture we will show that deciding the relational validity of such rules is *PSPACE*-complete.

Hoare logic, introduced by C. A. R. Hoare in 1969 [6], was the first formal system for the specification and verification of well-structured programs. This pioneering work initiated the field of program correctness and inspired dozens of technical articles [2, 1, 3]. For this achievement among others, Hoare received the Turing Award in 1980. A comprehensive introduction to Hoare logic can be found in [3].

Hoare logic uses a specialized syntax involving *partial correctness assertions* (PCAs) of the form $\{b\}\, p\, \{c\}$ and a deductive apparatus consisting of a system of specialized rules of inference. Under certain conditions, these rules are relatively complete [2]; essentially, the propositional fragment of the logic can be used to reduce partial correctness assertions to static assertions about the underlying domain of computation.

The propositional fragment of Hoare logic, called *propositional Hoare logic* (PHL), is subsumed by KAT. The reduction transforms PCAs to ordinary equations and the specialized rules of inference to equational implications (universal Horn formulas). The transformed rules are all derivable in KAT by pure equational reasoning. More generally, all Hoare-style inference rules of the form

$$\frac{\{b_1\}\, p_1\, \{c_1\},\ \ldots,\ \{b_n\}\, p_n\, \{c_n\}}{\{b\}\, p\, \{c\}} \tag{14.1}$$

that are valid over relational models are derivable in KAT; this is trivially false for PHL.

Encoding of While Programs and Partial Correctness Assertions

The encoding of the **while** programming constructs using the regular operators and tests originated with propositional dynamic logic (PDL) [4]. KAT is strictly less expressive than PDL, but is a simpler and purely equational, and is only *PSPACE*-complete, whereas PDL is *EXPTIME*-complete. In addition, PDL interpretations are restricted to relational models.

Halpern and Reif [5] prove *PSPACE*-completeness of strict deterministic PDL, but neither the upper nor the lower bound of the KAT *PSPACE*-completeness result follows from theirs. Not only are PDL semantics restricted to relational models, but the arguments of [5] depend on an additional nonalgebraic restriction: the relations interpreting atomic programs must be single-valued. Without this restriction, even if only **while** programs are allowed, PDL is *EXPTIME*-hard. In contrast, KAT imposes no such restrictions.

Hoare Logic

A common choice of programming language in Hoare logic is the language of **while** programs. The first-order version of this language contains a simple assignment $x := e$, conditional test **if** b **then** p **else** q, sequential composition $p\,;\,q$, and a looping construct **while** b **do** p.

The encoding of the **while** program constructs in KAT is as in PDL [4]:

$$p\,;\,q \overset{\text{def}}{=} pq \tag{14.2}$$

$$\textbf{if } b \textbf{ then } p \textbf{ else } q \overset{\text{def}}{=} bp + \overline{b}q \tag{14.3}$$

$$\textbf{while } b \textbf{ do } p \overset{\text{def}}{=} (bp)^*\overline{b}. \tag{14.4}$$

The basic assertion of Hoare logic is the *partial correctness assertion* (PCA)

$$\{b\}\,p\,\{c\}, \tag{14.5}$$

where b and c are formulas and p is a program. Intuitively, this statement asserts that whenever b holds before the execution of the program p, then if and when p halts, c is guaranteed to hold of the output state. It does not assert that p must halt.

For applications in program verification, the standard interpretation would be a Kleene algebra of binary relations on a set and the Boolean algebra of subsets of the identity relation.

Semantically, programs p in Hoare logic and dynamic logic (DL) are usually interpreted as binary input/output relations $p^{\mathcal{M}}$ on a domain of computation $\mathcal{M}$, and assertions are interpreted as subsets of $\mathcal{M}$ [2, 10]. The definition of the relation $p^{\mathcal{M}}$ is inductive on the structure of p; for example, $(p\,;\,q)^{\mathcal{M}} = p^{\mathcal{M}} \circ q^{\mathcal{M}}$, the ordinary relational composition of the relations corresponding to p and q. The meaning of the PCA (14.5) is the same as the meaning of the DL formula $b \to [p]\,c$, where $\to$ is ordinary propositional implication and the modal construct $[p]\,c$ is interpreted in the model $\mathcal{M}$ as the set of states s such that for all $(s,t) \in p^{\mathcal{M}}$, the output state t satisfies c.

Hoare logic provides a system of specialized rules for deriving valid PCAs, one rule for each programming construct. The verification process is inductive on the structure of programs. The traditional Hoare inference rules are:

Assignment rule

$$\{b\,[x/e]\,\}\; x := e\; \{b\} \tag{14.6}$$

Composition rule

$$\frac{\{b\}\; p\; \{c\}, \quad \{c\}\; q\; \{d\}}{\{b\}\; p\, ;\; q\; \{d\}} \tag{14.7}$$

Conditional rule

$$\frac{\{b \wedge c\}\; p\; \{d\}, \quad \{\neg b \wedge c\}\; q\; \{d\}}{\{c\}\; \textbf{if}\; b\; \textbf{then}\; p\; \textbf{else}\; q\; \{d\}} \tag{14.8}$$

While rule

$$\frac{\{b \wedge c\}\; p\; \{c\}}{\{c\}\; \textbf{while}\; b\; \textbf{do}\; p\; \{\neg b \wedge c\}} \tag{14.9}$$

Weakening rule

$$\frac{b' \to b, \quad \{b\}\; p\; \{c\}, \quad c \to c'}{\{b'\}\; p\; \{c'\}}. \tag{14.10}$$

Cook [2] showed that these rules are complete relative to first-order number theory when interpreted over the structure of arithmetic $\mathbb{N}$.

Propositional Hoare logic (PHL) consists of atomic proposition and program symbols, the usual propositional connectives, **while** program constructs, and PCAs built from these. Atomic programs are interpreted as binary relations on a set $\mathcal{M}$ and atomic propositions are interpreted as subsets of $\mathcal{M}$. The deduction system of PHL consists of the composition, conditional, while, and weakening rules (14.7)–(14.10) and propositional logic. The assignment rule (14.6) is omitted, since there is no first-order relational structure over which to interpret program variables; in practice, its role is played by PCAs over atomic programs that are postulated as assumptions.

In PHL, we are concerned with the problem of determining the validity of rules of the form

$$\frac{\{b_1\}\; p_1\; \{c_1\},\; \ldots,\; \{b_n\}\; p_n\; \{c_n\}}{\{b\}\; p\; \{c\}} \tag{14.11}$$

over relational interpretations. The premises $\{b_i\}\; p_i\; \{c_i\}$ take the place of the assignment rule (14.6) and are an essential part of the formulation.

Encoding Hoare Logic in KAT

The propositional Hoare rules can be derived as theorems of KAT. The PCA $\{b\}\ p\ \{c\}$ is encoded in KAT by the equation

$$b p \overline{c} \;=\; 0. \tag{14.12}$$

Intuitively, this says that the program p with preguard b and postguard $\overline{c}$ has no halting execution. An equivalent formulation is

$$bp \;=\; bpc, \tag{14.13}$$

which says intuitively that testing c after executing bp is always redundant.

The equivalence of (14.12) and (14.13) can be argued easily in KAT. This equivalence was observed by Manes and Arbib [9] in a different context. Assuming (14.12),

$$\begin{aligned} bp &= bp(c+\overline{c}) && \text{by the axiom } a1 = a \text{ and Boolean algebra} \\ &= bpc + bp\overline{c} && \text{by distributivity} \\ &= bpc && \text{by (14.12) and the axiom } a+0=a. \end{aligned}$$

Conversely, assuming (14.13),

$$\begin{aligned} bp\overline{c} &= bpc\overline{c} && \text{by (14.13)} \\ &= bp0 && \text{by associativity and Boolean algebra} \\ &= 0 && \text{by the axiom } a0 = 0. \end{aligned}$$

The equation (14.13) is equivalent to the inequality $bp \le bpc$, since the reverse inequality is a theorem of KAT; it follows immediately from the axiom $c \le 1$ of Boolean algebra and monotonicity of multiplication.

Using the encoding of **while** programs (14.2)–(14.4) and (14.13), the Hoare rules (14.7)–(14.10) take the following form:

Composition rule:

$$bp = bpc \wedge cq = cqd \quad\rightarrow\quad bpq = bpqd \tag{14.14}$$

Conditional rule:

$$bcp = bcpd \wedge \overline{b}cq = \overline{b}cqd \quad\rightarrow\quad c(bp + \overline{b}q) = c(bp+\overline{b}q)d \tag{14.15}$$

While rule:

$$bcp = bcpc \quad \to \quad c(bp)^*\overline{b} = c(bp)^*\overline{b}\,\overline{b}c \tag{14.16}$$

Weakening rule:

$$b' \leq b \wedge bp = bpc \wedge c \leq c' \quad \to \quad b'p = b'pc'. \tag{14.17}$$

These implications are to be interpreted as universal Horn formulas; that is, the variables are implicitly universally quantified. To establish the adequacy of the translation, we show that (14.14)–(14.17) encoding the Hoare rules (14.7)–(14.10) are theorems of KAT.

Theorem 14.1 *The universal Horn formulas* (14.14)–(14.17) *are theorems of* KAT*.*

Proof. First we derive (14.14). Assuming the premises

$$\begin{aligned} bp &= bpc && (14.18)\\ cq &= cqd, && (14.19) \end{aligned}$$

we have

$$\begin{aligned} bpq &= bpcq && \text{by (14.18)}\\ &= bpcqd && \text{by (14.19)}\\ &= bpqd && \text{by (14.18).} \end{aligned}$$

Thus the implication (14.14) holds.

For (14.15), assume the premises

$$\begin{aligned} bcp &= bcpd && (14.20)\\ \overline{b}cq &= \overline{b}cqd, && (14.21) \end{aligned}$$

Then

$$\begin{aligned} c(bp + \overline{b}q) &= cbp + c\overline{b}q && \text{by distributivity}\\ &= bcp + \overline{b}cq && \text{by commutativity of tests}\\ &= bcpd + \overline{b}cqd && \text{by (14.20) and (14.21)}\\ &= cbpd + c\overline{b}qd && \text{by commutativity of tests}\\ &= c(bp + \overline{b}q)d && \text{by distributivity.} \end{aligned}$$

For (14.16), by trivial simplifications it suffices to show

$$cbp \leq cbpc \quad \to \quad c(bp)^* \leq c(bp)^*c.$$

Assume

$$cbp \quad \leq \quad cbpc. \tag{14.22}$$

By an axiom of KA, we need only show

$$c + c(bp)^*cbp \quad \leq \quad c(bp)^*c.$$

But

$$\begin{array}{rcll} c + c(bp)^*cbp & \leq & c + c(bp)^*cbpc & \text{by (14.22) and monotonicity} \\ & \leq & c1c + c(bp)^*cbpc & \text{by Boolean algebra} \\ & \leq & c(1 + (bp)^*cbp)c & \text{by distributivity} \\ & \leq & c(1 + (bp)^*bp)c & \text{by monotonicity} \\ & \leq & c(bp)^*c & \text{by unwinding.} \end{array}$$

Finally, for (14.17), we can rewrite the rule as

$$b' \leq b \wedge bp\overline{c} = 0 \wedge \overline{c}' \leq \overline{c} \quad \rightarrow \quad b'p\overline{c}' = 0,$$

which follows immediately from the monotonicity of multiplication. □

References

[1] E. M. Clarke, S. M. German, and J. Y. Halpern. Effective axiomatizations of Hoare logics. *J. Assoc. Comput. Mach.*, 30:612–636, 1983.

[2] S. A. Cook. Soundness and completeness of an axiom system for program verification. *SIAM J. Comput.*, 7(1):70–90, February 1978.

[3] Patrick Cousot. Methods and logics for proving programs. In J. van Leeuwen, editor, *Handbood of Theoretical Computer Science*, volume B, pages 841–993. Elsevier, Amsterdam, 1990.

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

[5] J. Y. Halpern and J. H. Reif. The propositional dynamic logic of deterministic, well-structured programs. *Theor. Comput. Sci.*, 27:127–165, 1983.

[6] C. A. R. Hoare. An axiomatic basis for computer programming. *Comm. Assoc. Comput. Mach.*, 12:576–80, 1969.

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

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

[9] E. G. Manes and M. A. Arbib. *Algebraic Approaches to Program Semantics.* Springer-Verlag, New York, 1986.

[10] V. R. Pratt. A practical decision method for propositional dynamic logic. In *Proc. 10th Symp. Theory of Comput.*, pages 326–337. ACM, 1978.