A Note on the Complexity of Propositional Hoare Logic

Ernie Cohen[1]
Telcordia Technologies
and
Dexter Kozen[2]
Cornell University

We provide a simpler alternative proof of the *PSPACE*-hardness of propositional Hoare logic (PHL) [Kozen 2000].

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.2.2 [**Algebraic Manipulation**]: Automatic Programming—*program verification*

General Terms: Design, Languages, Theory, Verification

Additional Key Words and Phrases: Hoare logic, verification, specification

Kozen has shown that propositional Hoare logic (PHL) is *PSPACE*-complete [Kozen 2000, Theorem 5.1]. The proof of *PSPACE*-hardness is by a direct encoding of a polynomial-space Turing machine. In this note we provide a simpler proof encoding the universality problem for nondeterministic finite automata, a well-known *PSPACE*-complete problem [Garey and Johnson 1979]. This construction is particularly interesting because it can be used to turn a symbolic model checker such as SMV [McMillan 1992] into an efficient checker for regular expression equivalence.

[1] Address: Telcordia Technologies, Inc., 445 South St., Morristown, NJ 07960, USA. Email: ernie@research.telcordia.com.
[2] Address: Department of Computer Science, Cornell University, Ithaca, NY 14853-7501, USA. Email: kozen@cs.cornell.edu. The support of the National Science Foundation under grant CCR-9708915 is gratefully acknowledged.

We follow the notation of [Kozen 2000]. The deduction rules of PHL consist of the usual composition, conditional, while, and weakening rules of Hoare logic, as well as the and- and or-rule

$$\frac{\{c\}\, p\, \{d\},\ c \in C}{\{\bigvee C\}\, p\, \{d\}} \qquad \frac{\{b\}\, p\, \{c\},\ c \in C}{\{b\}\, p\, \{\bigwedge C\}}$$

for any finite set C of propositions. The and- and or-rule are not part of the traditional formulation [Apt 1981] but are necessary for completeness [Kozen and Tiuryn 2000]. The assignment axiom is meaningless in PHL and is omitted. We are interested in the validity of rules of the form

$$\frac{\{b_1\}\, p_1\, \{c_1\}, \ldots, \{b_n\}\, p_n\, \{c_n\}}{\{b\}\, p\, \{c\}} \tag{1}$$

interpreted as universal Horn sentences over relational models.

We consider two related decision problems: given a rule of the form (1),

(i) is it relationally valid? That is, is it true in all relational models?

(ii) is it derivable in PHL?

The paper [Kozen 2000] considered problem (i) only. We show that both of these problems are *PSPACE*-hard by a single reduction from the universality problem for nondeterministic finite automata: given such an automaton M over input alphabet $\{0,1\}$ with states Q, nondeterministic transition function $\Delta : Q \times \{0,1\} \to 2^Q$, start states $S \subseteq Q$, and final states $F \subseteq Q$, does M accept all strings?

The reduction to PHL is as follows. Let a_u be an atomic proposition for each state $u \in Q$. Let b be another atomic proposition and let p be an atomic program. Let

$$\textsc{start} \stackrel{\text{def}}{=} \bigwedge_{u \in S} a_u \qquad \textsc{final} \stackrel{\text{def}}{=} \bigvee_{u \in F} a_u.$$

Consider the rule

$$\frac{\{a_u \wedge b\}\, p\, \{a_v\} \text{ for all } v \in \Delta(u,1), \quad \{a_u \wedge \neg b\}\, p\, \{a_v\} \text{ for all } v \in \Delta(u,0)}{\{\textsc{start}\}\ \textbf{while}\ \textsc{final}\ \textbf{do}\ p\, \{\textsc{false}\}} \tag{2}$$

Note that this rule is linear in the size of the description of the automaton.

The rule (2) encodes a pebbling algorithm that simulates the subset construction. Intuitively, a_u says that there is a pebble on state u, and b (respectively, $\neg b$) says that the next input symbol is 1 (respectively, 0). The program p says to place pebbles on *at least* all states reachable from a currently pebbled state under the next input symbol according to the transition rules of M. The formula START says that all start states are pebbled, and FINAL says that at least one final state is pebbled. Other subexpressions of (2) have the following intuitive meanings:

$\{a_u \wedge b\}\, p\, \{a_v\}$	"If state u is pebbled at time t, and if the next input symbol is 1, then there must be a pebble on state v at time $t+1$."
while FINAL **do** p	"Continue updating the pebble positions as long as there is a final state occupied by a pebble."

The formula (2) says intuitively that if all start states are initially pebbled, and if in each step the pebbles are moved such that *at least* those states that are reachable under the current input symbol from a currently pebbled state are pebbled in the next step, then there is always a pebble on a final state.

Now we proceed to the formal proof of the correctness of this construction.

THEOREM 1. *The following are equivalent:*

(i) *The rule (2) is relationally valid.*

(ii) *The rule (2) is derivable in* PHL.

(iii) *The automaton M accepts all strings.*

PROOF. We show (ii) $\Rightarrow$ (i) $\Rightarrow$ (iii) $\Rightarrow$ (ii). The first implication is immediate from the soundness of PHL over relational models.

For the second implication, let $x \in \{0,1\}^*$ be any input string. Build a relational model of PHL as follows: the elements are the prefixes of x; the formula b is true at y if $x = yz$ and the first symbol of z is 1; the formula a_u is true at y if $u \in \Delta(S, y)$, that is, if the state u is reachable under input string y from a start state of M; and the program p is the relation consisting of all pairs (y, z) for $|z| = |y| + 1$. An easy argument shows that all premises of (2) hold in this model. By (i), the conclusion holds, thus x satisfies FINAL, so there is a final state reachable from a start state of M under input string x.

Finally, for the third implication, we prove (2) in PHL. Let

$$\mathcal{R} \stackrel{\text{def}}{=} \{\Delta(S, x) \mid x \in \{0,1\}^*\} \qquad \varphi \stackrel{\text{def}}{=} \bigvee_{A \in \mathcal{R}} \bigwedge_{s \in A} a_s.$$

The set $\mathcal{R}$ is just the set of reachable states of the subset automaton. It follows in a straightforward way from the premises of (2) using the and-, or-, and weakening rules that φ is an invariant of p, or in other words $\{\varphi\}\, p\, \{\varphi\}$. Since START $\rightarrow \varphi$ and $\varphi \rightarrow$ FINAL, the latter being a consequence of (iii), the conclusion of (2) follows from the while rule and weakening. □

ACKNOWLEDGMENTS

We thank the editor, Krzysztof Apt, for his forbearance during the preparation of this note.

REFERENCES

APT, K. R. 1981. Ten years of Hoare's logic: a survey—part I. *ACM Trans. Programming Languages and Systems 3*, 431–483.

GAREY, M. R. AND JOHNSON, D. S. 1979. *Computers and Intractibility: A Guide to the Theory of NP-Completeness*. W.H. Freeman.

KOZEN, D. 2000. On Hoare logic and Kleene algebra with tests. *Trans. Computational Logic*. This issue.

KOZEN, D. AND TIURYN, J. 2000. On the completeness of propositional Hoare logic. In J. DESHARNAIS Ed., *Proc. 5th Int. Seminar Relational Methods in Computer Science (RelMiCS 2000)* (January 2000), pp. 195–202.

MCMILLAN, K. L. 1992. *Symbolic model checking—an approach to the state explosion problem*. Ph. D. thesis, School of Computer Science, Carnegie Mellon University.