CS 819B - Spring 2004

Bibliographic Notes on Hoare Logic

John C. Reynolds — April 1, 2004

Hoare logic is given a theoretical treatment by J. Loeckx et al. [1, Chapter 8], who deal with completeness issues as well as soundness; other books are better for examples of real proofs. A number of partial-correctness proofs, in a formalism similar to that used in this course, are given in Reynolds [2, Chapter 1].

Various total-correctness proofs are given by Dijkstra [3] and D. Gries [4], but in a formalism using guarded commands and weakest preconditions.

The original papers in this area are unusually clear and well worth reading. The use of assertions to specify and prove correctness of flowchart programs was developed independently by Naur [5] and Floyd [6], although the essential idea goes back to the origins of computer programming, in unpublished reports by Goldstine and von Neumann [7] and Turing [8]. Then Hoare [9] developed the partial-correctness system described in this course. (Arithmetic with unchecked overflow was also formalized in this paper.) Later, Hoare [10] gave an exemplary proof of a median-finding program which made it clear that program-proving could be integrated with top-down program construction. The soundness of Hoare's system was first proved by P. E. Lauer [11] (as summarized in Hoare and Lauer [12]). (A modern soundness proof is given by Reynolds [13].) The concept of weakest preconditions (and guarded commands) was introduced by Dijkstra [14].

Program-specification logics that accommodate partial functions were devised by H. Barringer, J. H. Cheng, and C. B. Jones [15] and Jones and C. A. Middelburg [16]. Surveys of this area include Cheng and Jones [17] and O. Owe [18].

The inference rule for array assignment was originally introduced by Hoare and Wirth [19] in their axiomatic definition of Pascal. An alternative rule, based on substitution for array applications, was devised by Gries and G. Levin [20] and is also described in Gries [4, Chapter 9]. Examples of specifications and (mostly semi-formal) proofs concerning arrays, including the mathematical concepts described in the class notes on arrays and procedures, are given in Reynolds [21], [2, Chapter 2].

A variety of completeness and incompleteness theorems for Hoare-like reasoning systems, including extensions to powerful procedure mechanisms, are given by E. M. Clarke [22, 23, 24].

A short history of the early development of the subject, with an extensive bibliography, has been given by Jones [25]. A review of theoretical aspects was given by K. Apt [26].

A well-balanced and thoroughly researched history of program verification and its automation is given in a recent book by Donald MacKenzie [27].

References

[1] Jacques Loeckx, Kurt Sieber, and Ryan D. Stansifer. *The Foundations of Program Verification*. Wiley, Chichester, England, second edition, 1987.

[2] John C. Reynolds. *The Craft of Programming*. Prentice-Hall International, London, 1981.

[3] Edsger W. Dijkstra. *A Discipline of Programming*. Prentice-Hall, Englewood Cliffs, New Jersey, 1976.

[4] David Gries. *The Science of Programming*. Springer-Verlag, New York, 1981.

[5] Peter Naur. Proof of algorithms by general snapshots. *BIT*, 6:310–316, 1966.

[6] Robert W. Floyd. Assigning meanings to programs. In J. T. Schwartz, editor, *Mathematical Aspects of Computer Science*, volume 19 of *Proceedings of Symposia in Applied Mathematics*, pages 19–32, Providence, Rhode Island, 1967. American Mathematical Society.

[7] H. H. Goldstine and John von Neumann. Planning and coding of problems for an electronic computing instrument. Part II, Vol. 1 of a Report prepared for the U. S. Army Ordinance Department, 1947.

[8] Alan M. Turing. Checking a large routine. In *Report of a Conference on High Speed Automatic Calculating Machines*, pages 67–69, Cambridge, England, 1950. Marked "Issued by the Laboratory with the cooperation of the ministry of supply".

[9] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580 and 583, October 1969.

[10] C. A. R. Hoare. Proof of a program: FIND. *Communications of the ACM*, 14(1):39–45, January 1971.

[11] Peter E. Lauer. *Consistent Formal Theories of the Semantics of Programming Languages*. Ph. D. dissertation, Queen's University of Belfast, Belfast, Northern Ireland, 1971. Also Technical Report TR 25.121, IBM Laboratory Vienna, Vienna, Austria, November 14, 1971.

[12] C. A. R. Hoare and Peter E. Lauer. Consistent and complementary formal theories of the semantics of programming languages. *Acta Informatica*, 3(2):135–53, 1974.

[13] John C. Reynolds. *Theories of Programming Languages*. Cambridge University Press, Cambridge, England, 1998.

[14] Edsger W. Dijkstra. Guarded commands, nondeterminacy and formal derivation of programs. *Communications of the ACM*, 18(8):453–457, August 1975.

[15] Howard Barringer, J. H. Cheng, and Cliff B. Jones. A logic covering undefinedness in program proofs. *Acta Informatica*, 21(3):251–269, October 1984.

[16] Cliff B. Jones and C. A. Middelburg. A typed logic of partial functions reconstructed classically. *Acta Informatica*, 31(5):399–430, August 1994.

[17] J. H. Cheng and Cliff B. Jones. On the usability of logics which handle partial functions. In Carroll Morgan and J. C. P. Woodcock, editors, *3rd Refinement Workshop*, Workshops in Computing, pages 51–69, London, 1991. Springer-Verlag.

[18] Olaf Owe. Partial logics reconsidered: A conservative approach. *Formal Aspects of Computing*, 5(3):208–223, 1993.

[19] C. A. R. Hoare and Niklaus Wirth. An axiomatic definition of the programming language PASCAL. *Acta Informatica*, 2(4):335–355, 1973.

[20] David Gries and Gary Levin. Assignment and procedure call proof rules. *ACM Transactions on Programming Languages and Systems*, 2(4):564–579, October 1980.

[21] John C. Reynolds. Reasoning about arrays. *Communications of the ACM*, 22(5):290–299, May 1979.

[22] Edmund Melson Clarke Jr. *Completeness and Incompleteness Theorems for Hoare-like Axiom Systems.* Ph. D. dissertation, Cornell University, Ithaca, New York, 1976.

[23] Edmund Melson Clarke Jr. Program invariants as fixed points. In *18th Annual Symposium on Foundations of Computer Science*, pages 18–29, Long Beach, California, 1977. IEEE Computer Society.

[24] Edmund Melson Clarke Jr. Programming language constructs for which it is impossible to obtain good Hoare axiom systems. *Journal of the ACM*, 26(1):129–147, January 1979.

[25] Cliff B. Jones. The early search for tractable ways of reasoning about programs. *IEEE Annals of the History of Computing*, 25(2):26–49, 2003.

[26] Krzysztof R. Apt. Ten years of hoare's logic: A survey—part 1. *ACM Transactions on Programming Languages and Systems*, 3(4):431–483, October 1981.

[27] Donald MacKenzie. *Mechanizing Proof: Computing, Risk, and Trust.* MIT Press, Cambridge, Massachusetts, 2001.