CHAPTER 15

Conclusions

"You shall know* the truth, and the truth shall make you free."
— John 8:32

"But now having been set free from sin, and having become slaves of God, you have your fruit to holiness, and the end, everlasting life."
— Romans 6:22

* "**know,** *ginosko* (ghin-*oce*-koe); Strong's #1097: To perceive, understand, recognize, gain knowledge, realize, come to know. *Ginosko* is the knowledge that has an inception, a progress, and an attainment. It is the recognition of truth by personal experience."
— The Spirit-Filled Life Bible, Thomas Nelson Publishers, 1991, page 1589.

We have presented in this dissertation a verification condition generator tool for proving programs totally correct. We have verified the VCG, proving it sound from a foundation of a structural operational semantics. From this operational semantics we derived an axiomatic semantics, as theorems whose soundness was established by proof. From these we proved the correctness of the VCG. The entire proof has been conducted within the HOL mechanical theorem proving environment, guaranteeing the soundness of the reasoning and the verification result.

As part of this process, we developed five program logics, three of which were fundamental new inventions in this work, namely the expression logic, the entrance logic, and the termination logic. These regularized the process of proving termination for a program with mutually recursive procedures, and formed a structure less *ad hoc* than previous proposals.

This work has now provided a tool which can substantially decrease the difficulty of proving programs correct. It does not eliminate that difficulty, and even the use of this tool requires training and expertise. However, it points the direction towards mechanical assistance of the proof process which we believe is essential to the practical realization of the dream of widespread program verification. Such tools must not only be powerful and efficient, but it is vital that they themselves be trustworthy, for the proofs constructed using those tools can be no more reliable than the tools themselves.

This trustworthiness is now demonstrated to be feasible, by the presentation of this VCG tool. We believe that the annotation structure described is not onerous, but reasonable and intuitive. It is extremely important that whatever structure is imposed aids, and does not obstruct, the creation process. We have attempted to craft the annotation structures described in this work to be simple and structurally well placed, so as to provide the maximum strength with the minimum constraint. Extending this work to new language features and styles will require new annotation and proof structures. We look forward to further developments for greater strength in days to come.

References

[AA78] Suad Alagić and Michael A. Arbib. *The Design of Well-Structured and Correct Programs.* Springer-Verlag, 1978.

[AdB90] Pierre America and Frank de Boer. Proving total correctness of recursive procedures. *Information and Computation*, 84(2):129–162, February 1990.

[Age91] Sten Agerholm. Mechanizing program verification in HOL. In M. Archer, J. J. Joyce, K. N. Levitt, and P. J. Windley, editors, *Proceedings of the 1991 International Workshop on the HOL Theorem Proving System and its Applications*, pages 208–222. IEEE Computer Society Press, August 1991.

[And86] Peter B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof.* Academic Press, 1986.

[AO91] Krzysztof R. Apt and Ernst-Rüdiger Olderog. *Verification of Sequential and Concurrent Programs.* Springer-Verlag, New York, 1991.

[Apt81] K. R. Apt. Ten years of hoare logic: A survey—part 1. *ACM Transactions on Programming Languages and Systems*, 3(4):431–483, 1981.

[BGG$^+$92] Richard Boulton, Andrew Gordon, Mike Gordon, John Harrison, John Herbert, and John Van Tassel. Experience with embedding hardware description languages in HOL. In V. Stavridou, T. F. Melham, and T. T. Boute, editors, *Theorem Provers in Circuit Design*, pages 129–156. Elsevier Science Publishers B.V. (North Holland), 1992.

[BM81] Robert S. Boyer and J Strother Moore. A verification condition generator for FORTRAN. In Robert S. Boyer and J Strother Moore, editors, *The Correctness Problem in Computer Science.* Academic Press, London, 1981.

[BM88] Robert S. Boyer and J Strother Moore. *A Computational Logic Handbook.* Academic Press, 1988.

[Chu40] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5(2):56–68, June 1940.

[CM92] Juanito Camilleri and Tom Melham. Reasoning with inductively defined relations in the HOL theorem prover. Technical Report 265, University of Cambridge Computer Laboratory, August 1992.

[Coo78] Stephen A. Cook. Soundness and completeness of an axiom system for program verification. *SIAM Journal on Computing*, 7(1):70–90, February 1978.

[Dah92] Ole-Johan Dahl. *Verifiable Programming.* Prentice Hall International Series in Computer Science. Prentice Hall, London, 1992.

[dB80] Jaco de Bakker. *Mathematical Theory of Program Correctness.* Prentice Hall International, London, 1980.

[Dij72] Edsger W. Dijkstra. Notes on structured programming. In O.-J. Dahl, E. W. Dijkstra, and C. A. R. Hoare, editors, *Structured Programming.* Academic Press, 1972.

[Dij76] E. W. Dijkstra. *A Discipline of Programming.* Prentice Hall, 1976.

[Flo67] R. W. Floyd. Assigning meanings to programs. In J. T. Schwartz, editor, *Mathematical Aspects of Computer Science, Proceedings of the American Mathematical Society Symposia in Applied Mathematics*, volume 19, pages 19–31, Providence, R.I., 1967. American Mathematical Society.

[Fra92] Nissim Francez. *Program Verification.* Addison-Wesley, Wokingham, England, 1992.

[GM93] Michael J. C. Gordon and Thomas F. Melham. *Introduction to HOL: A Theorem Proving Environment for Higher-Order Logic.* Cambridge University Press, Cambridge, 1993.

[Gor88] Michael J. C. Gordon. *Programming Language Theory and its Implementation.* Prentice Hall International Series in Computer Science. Prentice Hall, London, 1988.

[Gor89] Michael J. C. Gordon. Mechanizing programming logics in higher order logic. In P. A. Subrahmanyam and G. Birtwistle, editors, *Current Trends in Hardware Verification and Automated Theorem Proving*, pages 387–489. Springer-Verlag, New York, 1989.

[Gra87] D. Gray. A pedagogical verification condition generator. *The Computer Journal*, 30(3):239–248, June 1987.

[Gri81] David Gries. *The Science of Programming.* Springer-Verlag, 1981.

[Hen90] Matthew Hennessy. *The Semantics of Programming Languages.* Wiley, 1990.

[HM94] Peter V. Homeier and David F. Martin. Trustworthy tools for trustworthy programs: A verified verification condition generator. In Thomas F. Melham and Juanito Camilleri, editors, *Proceedings of the 1994 International Workshop on the HOL Theorem Proving System and its Applications*, pages 269–284. Springer-Verlag, September 1994. LNCS 859.

[Hoa69] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–581, October 1969.

[Hoa71] C. A. R. Hoare. Procedures and parameters: an axiomatic approach. In E. Engeler, editor, *Proceedings of Symposium on Semantics of Algorithmic Languages*, volume 188 of *Lecture Notes in Mathematics*, Berlin, 1971. Springer-Verlag.

[ILL75] Shigeru Igarashi, Ralph L. London, and David C. Luckham. Automatic program verification: A logical basis and its implementation. *Acta Informatica*, 4:145–182, 1975.

[Kau94] Matt Kaufmann. Combining an interpreter-based approach to software verification with verification condition generation. Technical Report 97, Computational Logic, Inc., April 1994.

[Lin93] H. Lin. A verification tool for value-passing processes. In *Proceedings of PSTV XIII, Liege, Belgium*, May 1993.

[Mel89] Thomas F. Melham. Automating recursive type definitions in higher-order logic. In G. Birtwistle and P. Subrahmanyam, editors, *Current Trends in Hardware Verification and Automated Theorem Proving*, pages 341–386. Springer-Verlag, 1989.

[Mel91] Thomas F. Melham. A package for inductive relation definitions in HOL. In M. Archer, J. J. Joyce, K. N. Levitt, and P. J. Windley, editors, *Proceedings of the 1991 International Workshop on the HOL*

Theorem Proving System and its Applications, pages 350–357. IEEE Computer Society Press, August 1991.

[Mel92] Thomas F. Melham. A mechanized theory of the π-calculus in HOL. Technical Report 244, University of Cambridge Computer Laboratory, January 1992.

[Nes93] Monica Nesi. Value-passing CCS in HOL. In Jeffrey J. Joyce and Carl-Johan H. Seger, editors, *Proceedings of the HUG'93 6th International Workshop on Higher Order Logic Theorem Proving and Its Applications*, pages 352–365. Springer-Verlag, August 1993. LNCS 780.

[PJ86] P. Pandya and M. Joseph. A structure-directed total correctness proof rule for recursive procedure calls. *The Computer Journal*, 29(6):531–537, 1986.

[Plo81] Gordon Plotkin. A structured approach to operational semantics. Technical Report DAIMI FN-19, Aarhus University Computer Science Department, September 1981.

[Rag73] Larry Calvin Ragland. A verified program verifier. Technical Report 18, Department of Computer Sciences, The University of Texas at Austin, Austin, May 1973.

[Sok77] Stefan Sokołowski. Total correctness for procedures. In J. Gruska, editor, *Proceedings, 6th Symposium on the Mathematical Foundations of Computer Science*, pages 475–483. Springer-Verlag, September 1977. LNCS 53.

[Sok84] Stefan Sokołowski. Partial correctness: The term-wise approach. *Science of Computer Programming*, 4:141–157, 1984.

[Sto88] A. Stoughton. Substitution revisited. *Theoretical Computer Science*, 59:317–325, 1988.

[ZSO$^+$93] Cui Zhang, Rob Shaw, Ronald A. Olsson, Karl Levitt, Myla Archer, Mark R. Heckman, and Gregory D. Benson. Mechanizing a programming logic for the concurrent programming language microSR in HOL. In Jeffrey J. Joyce and Carl-Johan H. Seger, editors, *Proceedings of the 6th International Workshop on Higher Order Logic Theorem Proving* *and its Applications*, number 780 in LNCS, pages 29–42. Springer-Verlag, August 1993.