Journal of Automated Reasoning manuscript No.
(will be inserted by the editor)

Computer Assisted Reasoning

A Festschrift for Michael J. C. Gordon

Richard Boulton · Joe Hurd · Konrad Slind

30th June 2009

Today's increasingly computer-based society is dependent on the correctness and reliability of crucial infrastructure, such as programming languages, compilers, networks, and microprocessors. One important way to achieve the required level of assurance is to use formal specification and proof, and tool support for this approach has steadily grown to the point where the specification and verification of important system infrastructure is now feasible.

To survey the state of the art and discuss future possibilities and challenges, a two day research meeting entitled *Tools and Techniques for Verification of System Infrastructure*[1] was held in March 2008 at the Royal Society in London. The event was held in honour of Prof. Michael J. C. Gordon FRS on the occasion of his 60th birthday, and we are pleased to dedicate this special issue of the Journal of Automated Reasoning to him, which contains a selection of papers that followed from the meeting.

Career Overview Mike Gordon's career has been characterized by ground-breaking research on formal semantics for programming languages and machine-assisted formal verification. This focus has given us a creative and wide-ranging body of work. In 1970, Mike took the undergraduate degree in Mathematics at Cambridge University as a student of Gonville and Caius college. In 1973, he obtained a Ph.D. (supervisor: Rod Burstall) in the Edinburgh Department of Machine Intelligence for a dissertation entitled *Evaluation and Denotation of Pure LISP Programs*.[2] Mike returned to Cambridge the following year to obtain a Diploma in Linguistics, and then spent a year in Stanford before taking up a post-doctoral fellowship in Edinburgh to work on the LCF project led by Robin Milner. Following a short period as an SRC Advanced Research Fellow at Edinburgh, Mike took up a lecturership in the Cambridge University Computer Laboratory in 1981. He has been there ever since, becoming a Reader in 1988, and Professor of Computer Assisted Reasoning in 1996. Mike was elected a Fellow of the Royal Society in 1994.

Icera Inc., 2520 The Quadrant, Aztec West, Bristol, BS32 4AQ, U.K. · Galois, Inc., 421 S.W. 6th Ave., Ste. 300, Portland OR 97204, U.S.A. · Rockwell Collins Advanced Technology Center, 400 Collins Road, N.E. Cedar Rapids, IA 52498, U.S.A.

[1] `http://www.ttvsi.org/`
[2] Examiners: David Park and Robin Milner.

Programming Language Semantics Mike's Ph.D. explored the semantics of Lisp using domain theory, which had recently been created by Dana Scott. Lisp's dynamic binding was a particular focus in this work [14]. Some years later Mike also published an accessible undergraduate semantics textbook which focused on denotational methods [15].

Edinburgh LCF In the mid 1970s Mike and Chris Wadsworth replaced Lockwood Morris and Malcolm Newey as research assistants to Robin Milner working on Edinburgh LCF, a system designed by Milner as a successor to his earlier Stanford LCF. The project resulted in the ML language [23], a hugely influential force in computer science, plus the LCF theorem prover. LCF was both the name of the interactive proof system and the name of the logic (Logic for Computable Functions). The original system is discussed in [24]; an enhanced version due to Larry Paulson is documented in [39]. These original systems have not been maintained, but a version of the LCF logic continues to be distributed as an instance of the Isabelle generic proof system [40].

Hardware Verification Mike's research focus then shifted to hardware verification. He originally expressed hardware in *LCF_LSM*, a modification of the LCF system which incorporated ideas from CCS [33]. Using this system, he specified and verified a simple general-purpose computer, which subsequently became known as Gordon's computer [18]. Mike abandoned LCF_LSM in favour of higher order logic (HOL) following discussions with Ben Moszkowski; the benefits of higher-order functions for hardware are discussed in [19]. The modelling style advocated in that paper, namely to formalize devices as predicates on streams has been highly successful.

Hardware verification via interactive theorem proving attracted a great deal of interest at this time, and much of the leading work was being performed by Mike and his group at Cambridge. The techniques developed by them were applied to more sophisticated examples; for example, by Graham Birtwistle and Brian Graham at Calgary [28]. There were applications to the Viper military microprocessor [6,7] as a joint project between Mike's colleague (and wife) Avra Cohn and the Royal Signals and Radar Establishment (RSRE). This generated some controversy about the scope and limits of hardware verification [31] following the publication of an influential paper in this journal [8].

More recently, as a joint project with Birtwistle at Leeds, a model of an ARM instruction set architecture was shown to be correctly implemented by a model of the ARM6 microarchitecture [10,11]. This has lead to new methods for reasoning about low level software running on accurate hardware models [34]. In a related thrust, deduction-based algorithms are used to synthesize low-level implementations directly from functions defined in logic [34,42].

Higher Order Logic Much of the work on hardware verification conducted by Mike and his colleagues was performed using the HOL system. Besides being users of HOL, Mike and his students were also developing HOL, in particular the *HOL88* system [27]. Initially applied solely in the domain of hardware verification, higher order logic has since been applied to formalization and proof in a wide variety of settings, including pure and applied mathematics, hardware, and software. The HOL88 system spawned a number of mature descendants, including ProofPower [2], HOL-4 [41], Isabelle/HOL [36], and HOL Light [29].

A pervasive attitude—advocated early on by Mike—in the implementation of all these systems is the so-called *LCF approach*: the use of derived rules of definition and inference, which reduce all reasoning to primitive inference steps provided by a small kernel [17]. The LCF approach is tremendously flexible, and methods for efficient LCF-style proof have emerged [5,30].

In order to gather together the users of HOL for discussions, Mike initiated the *HOL Users Group* (HUG) series of meetings. This evolved into *TPHOLs* (Theorem Proving in Higher Order Logics), which provides a venue for research on any aspect of theorem proving with a flavour of higher order logic. TPHOLs has enjoyed robust health: this year (2009) finds the twenty-second instance being held in Munich.[3]

Formalized Semantics and Language Embeddings Mike and his students have a long history of formalizing language syntax and semantics. For example, Mike's influential paper [16] showed how a logic for a programming language could be derived from its formal semantics. The semantics of hardware description languages has been an ongoing activity. In early work, ELLA, VHDL, and SILAGE were discussed in [3]; the simulation-cycle semantics of Verilog appears in [20]; and a HOL formalization of the industry-standard PSL language appears in [22]. The semantics of many other computer languages have been formalized by Mike's students, among them CCS [35], π-calculus [32], C [37], C++ [38], and Java [43].

External Interfaces Mike has been a leader in the area of integration of theorem provers, efficient formula representations, and logics. In a joint project with Alan Bundy at Edinburgh, the HOL-Clam system [4] integrated the proof-planning capabilities of the Clam system with HOL. Mike was a member of the PROSPER project [9], led by Tom Melham, which provided exchange mechanisms for logical terms, formulas, and theories and investigated the coordination of a wide variety of reasoning and symbolic analysis tools. Mike oversaw the integration of BDDs into HOL, leading to a number of papers [21,1] showing how model-checking algorithms could be hosted on an LCF-style core inside HOL. In recent work, Mike has been collaborating with the authors of ACL2 in order to derive and exploit a verified translation between ACL2 and HOL [26,25].

Teaching Mike has long been recognized as a leader in teaching formal methods. His undergraduate course notes *Specification and Verification* [12,13] have been refined over years, and provide a time-tested and detailed resource for teaching program logic and hardware verification. These notes are characteristically clear and extremely detailed; we can personally testify that they have been re-used in many institutions worldwide.

Students The following is an alphabetical list of Mike's graduated Ph.D. students:

Hasan Amjad	Jim Grundy	Michael Norrish
Richard Boulton	John Harrison	James Reynolds
Albert Camilleri	Joe Hurd	Mark Staples
Rachel Cardell-Oliver	Juliano Iyoda	Daryl Stewart
Victor Carreño	Jeff Joyce	Donald Syme
Francisco Corella	Tom Melham	John Van Tassel
Inder Dhingra	Magnus Myreen	Stuart Wray
Jon Fairbairn	Monica Nesi	

[3] From 2010 the conference will be known as *Interactive Theorem Proving* (ITP).

Finally For us, and many others, Mike has been an ongoing source of research ideas and encouragement. His advice is unfailingly good. As a Ph.D. advisor and mentor, he has provided guidance of all kinds to his students and many others. Certainly, his Hardware Verification Group (HVG)[4] has over the years been a welcoming home, both to his students and to the many visitors who have come to Cambridge in order to share in the joys of doing research with Mike. We hope it goes on for many years to come!

References

1. Hasan Amjad, *Programming a symbolic model checker in a fully expansive theorem prover*, Theorem Proving in Higher Order Logics (TPHOLs), 16th International Conference, Proceedings, LNCS, vol. 2758, Springer, 2003, pp. 171–187.
2. Rob Arthan and Roger Bishop-Jones, *Z in Hol in ProofPower*, FACS FACTS, March 2005, pp. 387–439.
3. R. Boulton, A. Gordon, M. Gordon, J. Harrison, J. Herbert, and J. Van Tassel, *Experience with embedding hardware description languages in HOL*, Proceedings of the IFIP TC10/WG 10.2 International Conference on Theorem Provers in Circuit Design: Theory, Practice and Experience (Nijmegen, The Netherlands) (V. Stavridou, T. F. Melham, and R. T. Boute, eds.), IFIP Transactions, vol. A-10, North-Holland/Elsevier, June 1992, pp. 129–156.
4. Richard Boulton, Konrad Slind, Alan Bundy, and Mike Gordon, *An interface between Clam and HOL*, Theorem Proving in Higher Order Logics, 11th International Conference, TPHOLs'98 (Canberra) (Jim Grundy and Malcolm Newey, eds.), Lecture Notes in Computer Science, no. 1479, Springer-Verlag, 1998, pp. 87–104.
5. Richard Boulton, *Efficiency in a Fully-Expansive Theorem Prover*, Ph.D thesis, Tech. Report 337, University of Cambridge Computer Laboratory, May 1994.
6. Avra Cohn, *A proof of correctness of the Viper microprocessor: The first level*, VLSI Specification, Verification, and Synthesis (Graham Birtwistle and P.A. Subrahmanyam, eds.), Kluwer Academic Publishers, 1988, pp. 27–71.
7. ———, *Correctness of the Viper block model: The second level*, Current Trends in Hardware Verification and Automated Theorem Proving (Graham Birtwistle and P.A. Subrahmanyam, eds.), Springer-Verlag, 1989, pp. 1–91.
8. ———, *The notion of proof in hardware verification*, Journal of Automated Reasoning **5** (1989), no. 2, 127–139.
9. Louise A. Dennis, Graham Collins, Michael Norrish, Richard Boulton, Konrad Slind, Graham Robinson, Mike Gordon, and Tom Melham, *The PROSPER toolkit*, Tools and Algorithms for the Construction and Analysis of Systems: 6th International Conference, TACAS 2000 (Susanne Graf and Michael Schwartzbach, eds.), Lecture Notes in Computer Science, vol. 1785, Springer, March 2000, pp. 78–92.
10. A. Fox, *A HOL specification of the ARM instruction set architecture*, Tech. Report 545, University of Cambridge Computer Laboratory, June 2001.
11. ———, *Formal verification of the ARM6 micro-architecture*, Tech. Report 548, University of Cambridge Computer Laboratory, November 2002.
12. M. J. C. Gordon, *Specification and verification I*, 2009, Course notes available from `http://www.cl.cam.ac.uk/~mjcg/Teaching/SpecVer1/SpecVer1.html`.
13. ———, *Specification and verification II*, 2009, Course notes available from `http://www.cl.cam.ac.uk/~mjcg/Teaching/SpecVer2/SpecVer2.html`.
14. ———, *Evaluation and denotation of pure LISP programs*, Ph.D. thesis, University of Edinburgh, 1973.
15. ———, *The denotational description of programming languages: an introduction*, Springer Verlag, 1979.
16. ———, *Mechanizing programming logics in higher order logic*, Current Trends in Hardware Verification and Automated Theorem Proving, Springer-Verlag, 1989, pp. 387–439.
17. ———, *Representing a Logic in the LCF Metalanguage*, Tools and Notions for Program Construction, ed. D. N'eel, Cambridge University Press, 1982, pp. 163–185.

[4] Now the Automated Reasoning Group (ARG), run jointly with Larry Paulson.

18. ———, *Proving a computer correct with the LCF_LSM hardware verification system*, Tech. Report 42, University of Cambridge Computer Laboratory, 1983.
19. ———, *Why Higher-Order Logic is a good formalism for specifying and verifying hardware*, Tech. Report 77, University of Cambridge Computer Laboratory, 1985.
20. ———, *The semantic challenge of Verilog HDL*, Proceedings, Tenth Annual IEEE Symposium on Logic in Computer Science (LICS), 1995, pp. 136–145.
21. ———, *Reachability programming in HOL98 using BDDs*, Theorem Proving in Higher Order Logics: 13th International Conference, TPHOLs 2000 (M. Aagaard and J. Harrison, eds.), Lecture Notes in Computer Science, vol. 1869, Springer-Verlag, 2000, pp. 179–196.
22. ———, *Validating the PSL/Sugar semantics using automated reasoning*, Formal Aspects of Computing **15** (2003), no. 4, 406–421.
23. M. Gordon, R. Milner, L. Morris, M. Newey, and C. Wadsworth, *A Metalanguage for interactive proof in LCF*, POPL '78: Proceedings of the 5th ACM SIGACT-SIGPLAN symposium on Principles of programming languages (New York, NY, USA), ACM, 1978, pp. 119–130.
24. Michael Gordon, Robin Milner, and Christopher Wadsworth, *Edinburgh LCF: A mechanised logic of computation*, Lecture Notes in Computer Science, vol. 78, Springer-Verlag, 1979.
25. Michael J.C. Gordon, Warren A. Hunt, Matt Kaufmann, and James Reynolds, *An embedding of the ACL2 logic in HOL*, Proceedings of ACL2 2006, ACM International Conference Proceeding Series, vol. 205, ACM Press, 2006, pp. 40–46.
26. Michael J.C. Gordon, James Reynolds, Warren A. Hunt, and Matt Kaufmann, *An integration of HOL and ACL2*, Proceedings of FMCAD 2006, IEEE Computer Society, 2006, pp. 153–160.
27. Mike Gordon and Tom Melham, *Introduction to HOL, a theorem proving environment for higher order logic*, Cambridge University Press, 1993.
28. Brian Graham, *The SECD microprocessor, a verification case study*, Kluwer Academic Publishers, Boston, 1992.
29. John Harrison, *HOL-Light: A tutorial introduction*, Proceedings of the First International Conference on Formal Methods in Computer-Aided Design (FMCAD'96), LNCS, vol. 1166, Springer Verlag, 1996, pp. 265–269.
30. Joe Hurd, *First-Order Proof Tactics in Higher-Order Logic Theorem Provers*, Proceedings of STRATA 2003, First International Workshop on Design and Application of Strategies/Tactics in Higher Order Logics, NASA Tech. Report CP-2003-212448.
31. Donald MacKenzie, *Mechanizing proof: computing, risk, and trust*, MIT Press, 2001.
32. T. F. Melham, *A mechanized theory of the Π-calculus in HOL*, Nordic Journal of Computing **1** (1994), no. 1, 50–76.
33. Robin Milner, *Communication and concurrency*, International Series in Computer Science, Prentice Hall, 1989.
34. M. Myreen and M. Gordon, *Hoare logic for realistically modelled machine code*, Proceedings of TACAS 2007, LNCS, vol. 4424, Springer, 2007.
35. Monica Nesi, *Formalising process calculi in higher order logic*, Ph.D. thesis, University of Cambridge Computer Laboratory, 1996.
36. Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel, *Isabelle/HOL — a proof assistant for Higher-Order Logic*, LNCS, vol. 2283, Springer, 2002.
37. Michael Norrish, *C formalised in HOL*, Ph.D. thesis, University of Cambridge Computer Laboratory, 1998, Tech. Report Number UCAM-CL-TR-453.
38. ———, *A formal semantics for C++*, Tech. report, NICTA, 2008, Available from `http://nicta.com.au/people/norrishm/attachments/bibliographies_and_papers/C-TR.pdf`.
39. Lawrence Paulson, *Logic and computation: Interactive proof with Cambridge LCF*, Cambridge University Press, 1987.
40. Franz Regensburger, *HOLCF: Higher order logic of computable functions*, Higher Order Logic Theorem Proving and Its Applications: 8th International Workshop, LNCS, vol. 971, Springer Verlag, 1995, pp. 293–307.
41. Konrad Slind and Michael Norrish, *A brief overview of HOL-4*, TPHOLs (Otmane Aït Mohamed, César Muñoz, and Sofiène Tahar, eds.), LNCS, vol. 5170, Springer, 2008, pp. 28–32.
42. Konrad Slind, Scott Owens, Juliano Iyoda, and Mike Gordon, *Proof producing synthesis of arithmetic and cryptographic hardware*, Formal Aspects of Computing **19** (2007), no. 3, 343–362.
43. Donald Syme, *Declarative theorem proving for operational semantics*, Ph.D. thesis, University of Cambridge Computer Laboratory, 1998.