Lawrence Charles Paulson FRS

Professor of Computational Logic, University of Cambridge
Computer Laboratory, 15 JJ Thomson Avenue, Cambridge CB3 0FD
E-Mail: lp15@cam.ac.uk *Tel*: (01223) 334623

Academic Appointments

2002– *Professor of Computational Logic*, Cambridge. Lecture courses include *Logic and Proof* and *Interactive Formal Verification*. Various committee and administrative roles.

1987– *Fellow* and *Director of Studies*, Clare College, Cambridge. Responsible for admitting and teaching Clare undergraduates in Computer Science. Served on many College committees.

1998–02 *Reader in Computational Logic*, Cambridge. Various teaching and administrative duties.

1993–98 *University Lecturer*, Cambridge. Lectured on the foundations of logic programming, software engineering and the λ-calculus. Chaired the Staff-Student Liaison Committee.

1983–93 *Assistant Director of Research*, Cambridge. Lectured on LISP, ML, functional programming, polymorphism, etc. Organised the Computer Laboratory's Open Days.

1982–83 *Research assistant* on Prof. Robin Milner's LCF project, University of Edinburgh. Conceived and implemented many subsystems of the HOL proof assistant.

Education

1977–81 *Stanford University*, Stanford, CA. PhD, Computer Science. For thesis, developed a system to produce compilers from denotational definitions. Advisor: Prof. John Hennessy.

1973–77 *California Institute of Technology*, Pasadena, CA. BS, Mathematics. Graduated first in class. Studied computing, logic and set theory; attended Prof. N. G. de Bruijn's Automath lectures. Won Motorola Project Award for building a 70-chip computer for Conway's Life.

Editorial Positions

Editor of *J. Automated Reasoning* and member of the Programme Committees of leading conferences (CADE, TPHOLs, IJCAR, etc.) over many years. Founding editor of *LMS J. Computation and Mathematics* (1998–2007). PC Chairman (with Matt Kaufmann) of the inaugural *Interactive Theorem Proving* (ITP) conference, 2010. Since 2010, a Trustee of the *Conference on Automated Deduction* (CADE). Since 2016, Chairman of the ITP Steering Committee. Served on various committees of the Royal Society.

Honours / Awards

2017 Herbrand Award for Distinguished Contributions to Automated Reasoning

2017 Fellow of the Royal Society

2008 ACM Fellow

2006 Distinguished Affiliated Professor, Technical University of Munich

2003 Pilkington Teaching Prize

Personal

Born 20 September 1955, Philadelphia, Pennsylvania; US/UK nationality.

Married, two children from first marriage. Some competence in German, Russian, Spanish.

Research Summary

My field is automated theorem proving, which seeks computational methods for proving theorems in formal logic. In collaboration with Tobias Nipkow of T. U. Munich, I have developed a generic theorem prover, Isabelle, which constructs proofs in a variety of logics through a novel combination of typed λ-calculus and unification. Much of my work concerns improving the productivity of interactive proof construction by incorporating the techniques of automatic theorem provers. Isabelle is used in research projects worldwide. In earlier work (1982–85, jointly with Michael Gordon and Gérard Huet), I made major contributions to another theorem prover, HOL, which became hugely influential by 1990.

I have used Isabelle to formalize a substantial part of axiomatic set theory, in particular Gödel's celebrated incompleteness theorems and his proof of the relative consistency of the axiom of choice. I have developed a flexible method for modelling and verifying cryptographic protocols, which I have applied to real-world protocols, including TLS (used on secure websites) and SET, a huge cryptographic protocol suite for electronic commerce. I have recently developed MetiTarski, an automatic theorem prover for real valued special functions (sin, cos, ln, etc.). I was co-PI for a £1,051,627 project (joint with Edinburgh) that continued to develop and apply MetiTarski. I have recently been awarded a €2.4 million grant to investigate the use of Isabelle in mathematics research.

I have served as Principal Investigator for eleven UK-funded (EPSRC) grants, and served as a partner in four ESPRIT projects. I served on the committee that designed Standard ML and wrote a highly successful textbook on it (now in its 2nd edition, and translated into two other languages). Since 2003 I have belonged to the EPSRC Peer Review College, whose task is to evaluate funding applications.

Research Students

My PhD students include David Wolfram (who finished in 1991), Andrew Gordon (1992), Martin Coen (1992), Michael Hinchey (1999), Florian Kammüller (1999), Jacques Fleuriot (1999), Clemens Ballarin (1999), Giampaolo Bella (2000), Michael Bond (2004), Hyun-Jin Choi (2005), Jia Meng (2005), Michael Compton (2008), Aaron Coble (2009), James Bridge (2010), Jean Martina (2011), Nikolai Sultana (2014), William Denman (2015), Zongyan Huang (2015), William Sonnex (2016), Wenda Li (2018), Kawin Worrasangasilpa, Chaitanya Mangla and Chelsea Edmonds. Gordon and Fleuriot received CPHC/BCS Distinguished Dissertation awards (in 1994 and 2000); Wenda Li was one of two "highly commended runners-up" in 2020.

Professional Memberships

Association for Automated Reasoning (AAR), Association for Computing Machinery (ACM), London Mathematical Society (LMS), Royal Society, University and College Union (UCU)

Invited Lectures

1. Compiler generation from denotational semantics. *In*: B. Lorho (editor), *Methods and Tools for Compiler Construction* (Cambridge, 1984), 219–251. Notes for an advanced course held at INRIA-Rocquencourt, France in December 1983.

2. Lessons learned from LCF. DDC Workshop on Formal Software Development: Combining Specification Methods, Nyborg, Denmark (1984).

3. Joint BCS/SERC Workshop on Program Specification and Verification, York (1983)

4. BCS Formal Aspects of Computing Science, London (1984)

5. Specification and Derivation of Programs, Marstrand, Sweden (1985)

6. IEE/BCS Colloquium on Theorem Provers in Theory and Practice, London (1987)

7. Experience with Isabelle: a generic theorem prover. COLOG-88: International Conf. in Computer Logic, Tallinn, Estonia, USSR (1988).

8. Applications of proof theory to Isabelle. Proof Theory Summer School, Leeds (1990).

9. Tool support for logics of programs. *In*: Manfred Broy (editor), *Mathematical Methods in Program Development (Summer School Marktoberdorf 1996)* (Springer, 1997), 461–498.

10. Strategic principles in the design of Isabelle, *In*: Bernhard Gramlich and Frank Pfenning (editors), *CADE-15 Workshop on Strategies in Automated Deduction* (1998), 11–17.

11. Security protocols and their correctness. *Automated Reasoning Workshop 1998.* St. Andrews, Scotland (1998)

12. Proving security protocols correct. *IEEE Symposium on Logic in Computer Science.* Trento, Italy (1999).

13. Getting started with Isabelle. EEF Foundations School on Deduction and Theorem Proving. Edinburgh, Scotland (2000).

14. SET cardholder registration: the secrecy proofs. *In*: Rajeev Goré, Alexander Leitsch and Tobias Nipkow (editors), *Automated Reasoning: First International Joint Conference. IJCAR*, Siena, Italy. (Springer, 2001), 5–12.

15. Verifying the SET protocol: overview. *In*: Ali Abdallah, Peter Ryan and Steve Schneider (editors), *Formal Aspects of Security*, London, England (2002).

16. Formalizing abstract mathematics: issues and progress. IJCAR Workshop on *Computer-Supported Mathematical Theory Development*. Cork, Ireland (2004).

17. Computational logic and the quest for greater automation. Inaugural lecture upon conferment of the title of Distinguished Affiliated Professor, Munich, Germany (2006).

18. Automated assistance for proof assistants. *Colloquium in Honour of Gérard Huet*, Paris, France (2007).

19. Automation for interactive proof: techniques, lessons and prospects. *Tools and Techniques for Verification of System Infrastructure*, Royal Society, London, UK (2008).

20. The relative consistency of the axiom of choice mechanized using Isabelle/ZF. *Computability in Europe 2008* special session on Formalising Mathematics and Extracting Algorithms from Proofs, Athens, Greece (2008).

21. Three years of experience with Sledgehammer, a practical link between automatic and interactive theorem provers. *PAAR-2010: Workshop on Practical Aspects of Automated Reasoning.* Edinburgh, Scotland, July 2010. (Talk given by Jasmin Christian Blanchette.)

22. LCF + logical frameworks = Isabelle (25 years later). *Milner Symposium.* Edinburgh, Scotland, April 2012.

23. Overcoming intractable complexity in an automatic theorem prover for real-valued functions. *CCA 2012: Ninth International Conference on Computability and Complexity in Analysis.* Cambridge, England, June 2012.

24. MetiTarski: past and future. *Interactive Theorem Proving — ITP 2012*, Princeton, New Jersey (Springer LNCS 7406, 2012), 1–10.

25. Isabelle/HOL tutorial: verifying functional programs and inductively defined sets. *13th International Conference on Relational and Algebraic Methods in Computer Science (RAMiCS 13).* Cambridge, England, September 2012.

26. MetiTarski's menagerie of cooperating systems. Plenary lecture for the FroCoS and Tableaux conferences. Nancy, France, September 2013.

27. Theorem proving and the real numbers: overview and challenges. Keynote lecture for *NASA Formal Methods*. Houston, Texas, April 2014.

28. Automated theorem proving for special functions: the next phase. Keynote lecture for SNC 2014 (*Symbolic-Numeric Computation*). Shanghai, China, July 2014.

29. The future of formalised mathematics. Keynote lecture for EACA 2016 (XV Encuentro de Álgebra Computacional y Aplicaciones). Logroño, Spain, June 2016.

30. Porting the HOL Light analysis library: Some lessons. Keynote lecture for CPP 2017 (Certified Programs and Proofs). Paris, France, January 2017.

31. Proof assistants: From symbolic logic to real mathematics? "Jacques Morgenstern" Colloquium, Inria Sophia-Antipolis, France, May 2017.

32. Proof assistants: From symbolic logic to real mathematics? *Big Proof* programme, Isaac Newton Institute for Mathematical Sciences, Cambridge, UK, July 2017.

33. Proof support for hybrid system analysis. *Logical Methods for Safety and Security of Software Systems (Summer School Marktoberdorf 2017)*.

34. Automatic theorem proving: impressions from the interactive world. Herbrand Award acceptance speech. *Conference on Automated Deduction (CADE-26).* Gothenburg, Sweden, August 2017.

35. A career in research: Mike Gordon and hardware verification. *Automated Reasoning Workshop*, Cambridge, UK, April 2018.

36. Formalising mathematics in simple type theory. *Big Proof* programme, International Centre for Mathematical Sciences, Edinburgh, UK, May 2019.

37. Machine learning and the formalisation of mathematics: research challenges. Keynote lecture for Conference on Artificial Intelligence and Theorem Proving (AITP). Aussois, France (virtually), September 2020.

Publications by Lawrence C. Paulson FRS

Software Contributions

My research has always involved writing software. Software is not usually refereed; instead, its significance can be inferred by its take-up by the research community.

CGSG/PSP For my PhD, I developed a software system to analyse a formal description of the syntax and semantics of a programming language and generate a compiler. The resulting compilers were extremely inefficient, but the process was automatic and driven by high-level specifications. I demonstrated it by compiling and executing a seven-page program written in a large subset of Pascal. My system is sometimes called CGSG (Compiler Generator for Semantic Grammars) or PSP (Paulson's Semantics Processor). These names were coined by various other researchers.

LCF and HOL I substantially rewrote Edinburgh LCF, an interactive theorem prover developed by Robin Milner and his colleagues. My contributions included an implementation of full predicate logic, a new simplifier, a comprehensive library of semi-automatic proof tactics, and a huge improvement in performance. Much of this code became part of M. J. C. Gordon's HOL system and is still in use today in major universities around the world.

Isabelle I developed Isabelle in 1986 and was its sole developer for years. Since 1990, Tobias Nipkow and his colleagues have taken an increasingly prominent role. My contributions include the use of higher-order unification, search tactics based on streams, proof automation, the formalisation of axiomatic set theory, the treatment of inductive definitions and recursive data types, and the inductive approach to verifying security protocols. Sledgehammer, which allows automatic theorem provers to generate Isabelle proofs, was developed in my group with my substantial direct involvement.

MetiTarski MetiTarski, an automatic theorem prover for real-valued special functions, integrates Joe Hurd's Metis prover and the decision procedure QEPCAD. The design and implementation is primarily my work, assisted by Behzad Akbarpour.

LEO-II LEO-II, an automatic theorem prover for higher-order logic, was developed at Cambridge by Christoph Benzmüller. I organised and managed this research project. LEO-II came first in the higher-order logic category of the 2010 CADE ATP System Competition, which is the world championship for automated theorem provers. (Isabelle came third.)

Books, Edited Works, etc

1. L. C. Paulson. *A Compiler Generator for Semantic Grammars*. PhD Thesis, Stanford University (1981).

2. L. C. Paulson. *Logic and Computation: Interactive proof with Cambridge LCF* (Cambridge University Press, 1987).

3. L. C. Paulson. *ML for the Working Programmer* (Cambridge University Press, 1991). 2nd edition, 1996.

4. L. C. Paulson. *Isabelle: A Generic Theorem Prover* (Springer LNCS 828, 1994).

5. L. C. Paulson (editor). Proceedings of the First Isabelle Users Workshop. Technical Report 379, Computer Lab (1995).

6. Tobias Nipkow, L. C. Paulson and Markus Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic* (Springer LNCS 2283, 2002).

7. Bernhard Beckert and Lawrence C. Paulson (Editors). Special Issue on Automated Reasoning with Analytic Tableaux and Related Methods. *J. Automated Reasoning* **38** (2007) 1–3.

8. Matt Kaufmann and L. C. Paulson (editors). *Interactive Theorem Proving.* First International Conference, ITP 2010, Edinburgh, UK (Springer LNCS 6172, 2010).

Refereed Articles

1. L. C. Paulson. A semantics-directed compiler generator. *Principles of Programming Languages* (1982), 224–233.

2. L. C. Paulson. A higher-order implementation of rewriting. *Sci. Computer Programming* **3** (1983), 119–149.

3. L. C. Paulson. Deriving structural induction in LCF. *In*: G. Kahn, D. B. MacQueen, G. Plotkin, editors, *Semantics of Data Types* (Springer, 1984), 197–214.

4. L. C. Paulson. Verifying the unification algorithm in LCF. *Sci. Computer Programming* **5** (1985), 143–170.

5. L. C. Paulson. Lessons learned from LCF: a survey of natural deduction proofs. *Computer J.* **28** (1985), 474–479.

6. L. C. Paulson. Proving termination of normalization functions for conditional expressions. *J. Automated Reasoning* **2** (1986), 63–74.

7. L. C. Paulson. Constructing recursion operators in Intuitionistic Type Theory. *J. Symbolic Computation* **2** (1986), 325–355.

8. L. C. Paulson. Natural deduction as higher-order resolution. *J. Logic Programming* **3** (1986), 237–258.

9. L. C. Paulson. Isabelle: The next seven hundred theorem provers (system abstract). *In*: E. Lusk and R. Overbeek (editors), *9th International Conf. on Automated Deduction* (Springer LNCS 310, 1988), 772–773.

10. L. C. Paulson. The foundation of a generic theorem prover. *J. Automated Reasoning* **5** (1989), 363–397.

11. L. C. Paulson. A formulation of the simple theory of types (for Isabelle). *In*: P. Martin-Löf & G. Mints (editors), *COLOG-88: International Conf. in Computer Logic* (Springer, 1990), 246–274.

12. L. C. Paulson. Isabelle: The next 700 theorem provers. *In*: P. Odifreddi (editor), *Logic and Computer Science* (Academic Press, 1990), 361–386.

13. L. C. Paulson and Andrew W. Smith. Logic programming, functional programming, and inductive definitions. *In*: P. Schroeder-Heister (editor), *Extensions of Logic Programming* (Springer, 1991), 283–310.

14. L. C. Paulson. Designing a theorem prover. *In*: S. Abramsky, D. M. Gabbay, T. S. E. Maibaum (editors), *Handbook of Logic in Computer Science*, Vol II (Oxford, 1992), 415–475.

15. Tobias Nipkow and L. C. Paulson. Isabelle-91. *In*: D. Kapur (editor), *11th International Conf. on Automated Deduction* (Springer, 1992), 673–676.

16. L. C. Paulson. Set theory for verification: I. From foundations to functions. *J. Automated Reasoning* **11** (1993), 353–389.

17. L. C. Paulson. A fixedpoint approach to implementing (co)inductive definitions. *In*: A. Bundy (editor), *12th International Conf. on Automated Deduction* (Springer LNAI 814, 1994), 148–161. Longer version available as Report 320, Computer Lab (1993).

18. L. C. Paulson. Set theory for verification: II. Induction and recursion. *J. Automated Reasoning* 15 (1995), 167–215.

19. L. C. Paulson. A concrete final coalgebra theorem for ZF set theory. *In*: P. Dybjer, B. Nordstrom and J. Smith (editors), *TYPES '94: Types for Proofs and Programs* (Springer LNCS 996, 1995), 120–139.

20. L. C. Paulson and Krzysztof Grabczewski. Mechanizing set theory: cardinal arithmetic and the axiom of choice, *J. Automated Reasoning* **17** (1996), 291–323.

21. L. C. Paulson. Mechanizing coinduction and corecursion in higher-order logic. *J. Logic and Computation* 7 (1997), 175–204.

22. L. C. Paulson. Generic automatic proof tools, *In*: Robert Veroff (editor), *Automated Reasoning and its Applications: Essays in Honor of Larry Wos* (MIT Press, 1997), Chapter 3.

23. L. C. Paulson. Proving properties of security protocols by induction. *10th Computer Security Foundations Workshop* (IEEE Computer Society Press, 1997), 70–83.

24. L. C. Paulson. Mechanized proofs for a recursive authentication protocol. *10th Computer Security Foundations Workshop* (IEEE Computer Society Press, 1997), 84–95.

25. Giampaolo Bella and L. C. Paulson. Using Isabelle to prove properties of the Kerberos Authentication System. *DIMACS Workshop on Design and Formal Verification of Security Protocols*, September 3–5, 1997.

26. L. C. Paulson. The inductive approach to verifying cryptographic protocols. *J. Computer Security* **6** (1998), 85–128.

27. Giampaolo Bella and L. C. Paulson. Mechanising BAN Kerberos by the inductive method. *In*: Alan J. Hu and Moshe Y. Vardi (editors), *Computer-Aided Verification: CAV '98* (Springer LNCS 1427, 1998), 416–427.

28. Jacques Fleuriot and L. C. Paulson. A combination of nonstandard analysis and geometry theorem proving, with application to Newton's Principia. *In*: Claude Kirchner and Hélène Kirchner (editors), *15th International Conf. on Automated Deduction: CADE-15* (1998), 3–16.

29. L. C. Paulson. A generic tableau prover and its integration with Isabelle (preliminary version). CADE-15 Workshop on *Integration of Deductive Systems* (1998), 51–60. Also Report 441, Computer Lab (1998).

30. Clemens Ballarin and L. C. Paulson. Reasoning about coding theory: the benefits we get from computer algebra. *In*: Jacques Calmet and Jan Plaza (editors), *Artificial Intelligence and Symbolic Computation: AISC '98* (Springer LNAI 1476, 1998), 55–66.

31. Giampaolo Bella and L. C. Paulson. Kerberos version IV: inductive analysis of the secrecy goals. *In*: Jean-Jacques Quisquater et al. (editors), *Computer Security — ESORICS 98* (Springer LNCS 1485, 1998), 361–375.

32. Jacques Fleuriot and L. C. Paulson. Proving Newton's *Propositio Kepleriana* Using Geometry and Nonstandard Analysis in Isabelle. *In*: Xiao-Shan Gao, Dongming Wang and Lu Yang (editors), *Automated Deduction in Geometry, Second International Workshop, ADG'98* (Springer LNCS 1669, published 1999), 47–66.

33. Florian Kammüller and L. C. Paulson. A formal proof of Sylow's theorem: An experiment in abstract algebra with Isabelle HOL. *J. Automated Reasoning* **23** (1999), 235–264.

34. L. C. Paulson. Inductive analysis of the Internet protocol TLS. *ACM Trans. Information and System Security* **2**:3 (1999), 332–351.

35. L. C. Paulson. Final coalgebras as greatest fixed points in ZF set theory. *Mathematical Structures in Computer Science* **9** (1999), 545–567.

36. Clemens Ballarin and L. C. Paulson. A pragmatic approach to extending provers by computer algebra — with applications to coding theory. *Fundamenta Informaticæ* **39** (1999), 1–20.

37. Leslie Lamport and L. C. Paulson. Should your specification language be typed? *ACM Trans. Programming Languages and Systems* **21**:3 (1999), 502–526.

38. Florian Kammüller, Markus Wenzel and L. C. Paulson. Locales: A sectioning concept for Isabelle. *In*: Yves Bertot et al. (editors), *Theorem Proving in Higher Order Logics* (Springer LNCS 1690, 1999), 149–165.

39. L. C. Paulson. A generic tableau prover and its integration with Isabelle. *J. Universal Computer Science* **5**:3 (1999), at URL `http://www.iicm.edu/jucs_5_3/a_generic_tableau_prover`

40. L. C. Paulson. Mechanizing UNITY in Isabelle. *ACM Trans. on Computational Logic* **1**:1 (2000), 3–32.

41. L. C. Paulson. A fixedpoint approach to (co)inductive and (co)datatype definitions. *In*: Gordon Plotkin, Colin Stirling, and Mads Tofte (editors), *Proof, Language, and Interaction: Essays in Honour of Robin Milner* (MIT Press, 2000), 187–211.

42. Giampaolo Bella, Fabio Massacci, L. C. Paulson and Piero Tramontano. Formal verification of Cardholder Registration in SET. *In*: F. Cuppens et al. (editors), *Computer Security — ESORICS 2000* (Springer LNCS 1895, 2000), 159–174.

43. Jacques Fleuriot and L. C. Paulson. Mechanising nonstandard real analysis. *LMS J. Computation and Mathematics* **3** (2000), 140–190. At URL `http://www.lms.ac.uk/jcm/3/lms1999-027/`.

44. L. C. Paulson. Relations between secrets: two formal analyses of the Yahalom protocol. *J. Computer Security* **9**:3 (2001), 197–216.

45. L. C. Paulson. A simple formalization and proof for the mutilated chess board. *Logic Journal of the IGPL* **9**:3 (2001), 499–509.

46. Giampaolo Bella and L. C. Paulson. Mechanical proofs about a non-repudiation protocol. *In*: Richard J. Boulton and Paul B. Jackson (editors), *Theorem Proving in Higher Order Logics* (Springer LNCS 2152, 2001), 91–104.

47. L. C. Paulson. Mechanizing a theory of program composition for UNITY. *ACM Trans. on Programming Languages and Systems* **25**:5 (2001), 626–656.

48. Sidi O. Ehmety and L. C. Paulson. Program composition in Isabelle/UNITY. *In*: Michel Charpentier and Beverly Sanders (editors), Workshop on *Formal Methods for Parallel Programming: Theory and Application*, held at the *Parallel and Distributed Processing Symposium* (IEEE Press, 2002).

49. L. C. Paulson. The reflection theorem: a study in meta-theoretic reasoning. *In*: Andrei Voronkov (editor), *18th International Conf. on Automated Deduction: CADE-18* (Springer LNAI 2392, 2002), 377–391.

50. Giampaolo Bella, Fabio Massacci and L. C. Paulson. The verification of an industrial payment protocol: The SET purchase phase. *In*: Vijay Atluri (editor), *9th ACM Conference on Computer and Communications Security* (ACM Press, 2002), 12–20.

51. Giampaolo Bella, Fabio Massacci and L. C. Paulson. Verifying the SET registration protocols. *IEEE Journal on Selected Areas in Communications* **21**:1 (2003), 77–87.

52. Giampaolo Bella, Cristiano Longo and L. C. Paulson. Verifying second-level security protocols. *In*: David Basin and Burkhart Wolff (editors), *Theorem Proving in Higher Order Logics* (Springer LNCS 2758, 2003), 352–366.

53. L. C. Paulson. The relative consistency of the axiom of choice—mechanized using Isabelle/ZF. *LMS J. Computation and Mathematics* **6** (2003), 198–248. At URL `http://www.lms.ac.uk/jcm/6/lms2003-001/`.

54. Jia Meng and L. C. Paulson. Experiments on supporting interactive proof using resolution. *In*: David Basin and Michaël Rusinowitch (editors), *Automated Reasoning: Second International Joint Conference. IJCAR*. (Springer, 2004), 372–384.

55. L. C. Paulson. Organizing numerical theories using axiomatic type classes. *J. Automated Reasoning* **33**:1 (2004), 29–49.

56. Giampaolo Bella, Fabio Massacci and L. C. Paulson. An overview of the verification of SET. *International Journal of Information Security* **4** (2005), 17–28.

57. Sidi O. Ehmety and L. C. Paulson. Mechanizing compositional reasoning for concurrent systems: some lessons. *Formal Aspects of Computing* **17** (2005), 58–68.

58. Tobias Nipkow and L. C. Paulson. Proof pearl: defining functions over finite sets. *In*: Joe Hurd and Tom Melham (editors), *Theorem Proving in Higher Order Logics* (Springer LNCS 3603, 2005), 385–396.

59. L. C. Paulson. Defining functions on equivalence classes. *ACM Trans. on Computational Logic* 7:4 (2006), 658–675.

60. Giampaolo Bella, Fabio Massacci and L. C. Paulson. Verifying the SET purchase protocols. *J. Automated Reasoning* **36** (2006), 5–37.

61. Jia Meng, Claire Quigley and L. C. Paulson. Automation for interactive proof: first prototype. *Information and Computation* **204**:10 (2006), 1575–1596.

62. Giampaolo Bella and L. C. Paulson. Accountability protocols: formalized and verified. *ACM Trans. Information and System Security* **9**:2 (2006), 138–161.

63. Jia Meng and L. C. Paulson. Lightweight relevance filtering for machine-generated resolution problems. *In*: Geoff Sutcliffe, Renate Schmidt and Stephan Schulz (editors), *ESCoR: Empirically Successful Computerized Reasoning* (CEUR Workshop Proceedings, Vol. 192, 2006), 53–69.

64. Jia Meng and L. C. Paulson. Translating higher-order problems to first-order clauses. *In*: Geoff Sutcliffe, Renate Schmidt and Stephan Schulz (editors), *ESCoR: Empirically Successful Computerized Reasoning* (CEUR Workshop Proceedings, Vol. 192, 2006), 70–80.

65. Behzad Akbarpour and L. C. Paulson. Towards automatic proofs of inequalities involving elementary functions. *In*: Byron Cook and Roberto Sebastiani (editors), *PDPAR 2006: Pragmatics of Decision Procedures in Automated Reasoning*, 27–37.

66. Jia Meng, Lawrence C. Paulson and Gerwin Klein. A Termination Checker for Isabelle Hoare logic. *In*: Bernhard Beckert (editor), *4th International Verification Workshop, VERIFY '07* (CEUR Workshop Proceedings, Vol. 259, 2007), 104–118.

67. L. C. Paulson and Kong Woei Susanto. Source-level proof reconstruction for interactive theorem proving. *In*: Klaus Schneider and Jens Brandt (editors), *Theorem Proving in Higher Order Logics* (Springer LNCS 4732, 2007), 232–245.

68. Behzad Akbarpour and Lawrence C. Paulson. Extending a resolution prover for inequalities on elementary functions. *In*: Nachum Dershowitz and Andrei Voronkov (editors), *Logic for Programming, Artificial Intelligence, and Reasoning—LPAR 2007* (Springer LNCS 4790, 2007), 47–61.

69. Jia Meng and L. C. Paulson. Translating higher-order clauses to first-order clauses. *J. Automated Reasoning* **40**:1 (2008), 35–60.

70. Behzad Akbarpour and Lawrence C. Paulson. MetiTarski: an automatic prover for the elementary functions. *In*: Serge Autexier et al. (editors), *Intelligent Computer Mathematics* (Springer LNAI 5144, 2008), 217–231.

71. Christoph Benzmüller, L. C. Paulson, Frank Theiss and Arnaud Fietzke. LEO-II — a cooperative automatic theorem prover for classical higher-order logic. *In*: Alessandro Armando, Peter Baumgartner, Gilles Dowek (editors), *Automated Reasoning—4th International Joint Conference, IJCAR 2004* (Springer LNCS 5195, 2008), 162–170.

72. Jia Meng and L. C. Paulson. Lightweight relevance filtering for machine-generated resolution problems. *J. Applied Logic* **7**:1 (2009), 41–57.

73. Behzad Akbarpour and Lawrence C. Paulson. Applications of MetiTarski in the verification of control and hybrid systems. *In*: Rupak Majumdar and Paulo Tabuada (editors), *Hybrid Systems: Computation and Control* (Springer LNCS 5469, 2009), 1–15.

74. William Denman, Behzad Akbarpour, Sofiène Tahar, Mohamed H. Zaki and Lawrence C. Paulson. Formal verification of analog designs using MetiTarski. *In*: Armin Biere and Carl Pixley (editors), *Formal Methods in Computer Aided Design* (IEEE, 2009), 93–100.

75. Behzad Akbarpour and Lawrence C. Paulson. MetiTarski: An automatic theorem prover for real-valued special functions. *J. Automated Reasoning* **44**:3 (2010), 175–205.

76. Rajeev Narayanan, Behzad Akbarpour, Mohamed H. Zaki, Sofiène Tahar and L. C. Paulson. Formal verification of analog circuits in the presence of noise and process variation. In: *DATE 2010: Design, Automation and Test in Europe*, 1309–1312.

77. Christoph Benzmüller and Lawrence C. Paulson. Multimodal and intuitionistic logics in simple type theory. *Logic Journal of IGPL* **18**:6 (2010), 881–892. Doi: 10.1093/jigpal/jzp080.

78. Jasmin Christian Blanchette, Sascha Böhme and L. C. Paulson. Extending Sledgehammer with SMT solvers. *In*: Nikolaj Bjørner and Viorica Sofronie-Stokkermans (editors), *23rd International Conf. on Automated Deduction: CADE-23* (Springer LNAI 6803, 2011), 116–130.

79. Grant O. Passmore, Lawrence C. Paulson and Leonardo de Moura. Real algebraic strategies for Metitarski proofs. *In*: Johan Jeuring (editor), *Intelligent Computer Mathematics* (Springer LNAI 7362, 2012), 358–370.

80. Christoph Benzmüller and Lawrence C. Paulson. Quantified multimodal logics in simple type theory. *Logica Universalis* **7**:1 (2013), 7–20.

81. James P. Bridge and Lawrence C. Paulson. Case splitting in an automatic theorem prover for real-valued special functions. *J. Automated Reasoning* **50**:1 (2013), 99–117.

82. Nik Sultana, Jasmin Christian Blanchette and Lawrence C. Paulson. LEO-II and Satallax on the Sledgehammer test bench. *J. Applied Logic* **11**:1 (2013), 91–102.

83. Jasmin Christian Blanchette, Sascha Böhme and L. C. Paulson. Extending Sledgehammer with SMT solvers. *J. Automated Reasoning* **51**:1 (2013), 109–128.

84. Jean E. Martina and Lawrence C. Paulson. Verifying multicast-based security protocols using the inductive method. *SAC'13: 28th Annual ACM Symposium on Applied Computing* (ACM, 2013), 1824–1829.

85. James P. Bridge, Sean B. Holden and L. C. Paulson. Machine learning for first-order theorem proving: learning to select a good heuristic. *J. Automated Reasoning* **53**:2 (2014), 141–172.

86. L. C. Paulson. A machine-assisted proof of Gödel's incompleteness theorems for the theory of hereditarily finite sets. *Review of Symbolic Logic* **7**:3 (2014), 484–498.

87. Paul Jackson, Andrew Sogokon, James Bridge, L. C. Paulson. Verifying hybrid systems involving transcendental functions. *In*: Julia M. Badger and Kristin Y. Rozier (editors), *NASA Formal Methods* (Springer LNCS 8430, 2014), 188–202.

88. Zongyan Huang, Matthew England, David Wilson, James H. Davenport, Lawrence C. Paulson and James Bridge. Applying machine learning to the problem of choosing a heuristic to select the variable ordering for cylindrical algebraic decomposition. *In*: Stephen Watt (editor), *Intelligent Computer Mathematics* (Springer LNAI 8543, 2014), 92–107.

89. Zongyan Huang, Matthew England, David Wilson, James H. Davenport and L. C. Paulson. A comparison of three heuristics to choose the variable ordering for CAD. In: *ISSAC 2014: International Symposium on Symbolic and Algebraic Computation* (abstract for poster session). Also *ACM Communications in Computer Algebra* **48**:3 (September 2014), 121–123.

90. Jean E. Martina and L. C. Paulson. Verifying multicast-based security protocols using the inductive method. *International Journal of Information Security* **14**:2 (2015), 187–204.

91. L. C. Paulson. A mechanised proof of Gödel's incompleteness theorems using Nominal Isabelle. *J. Automated Reasoning* **55**:1 (2015), 1–37.

92. L. C. Paulson. A formalisation of finite automata using hereditarily finite sets. *In*: Amy P. Felty and Aart Middeldorp (editors), *25th International Conf. on Automated Deduction: CADE-25* (Springer LNAI 9195, 2015), 231–245.

93. Christoph Benzmüller, Nik Sultana, L. C. Paulson and Frank Theiss. The higher-order prover Leo-II. *J. Automated Reasoning* **55**:4 (2015), 389–404.

94. Nik Sultana, Christoph Benzmüller and L. C. Paulson. Proofs and Reconstructions. *In*: Carsten Lutz and Silvio Ranise (editors), *International Symposium on Frontiers of Combining Systems — FroCoS'15* (Springer LNAI 9322, 2015), 256–274.

95. Wenda Li and Lawrence C. Paulson. A modular, efficient formalisation of real algebraic numbers. *In*: Jeremy Avigad and Adam Chlipala (editors), *CPP 2016 — ACM SIGPLAN Conference on Certified Programs and Proofs* (2016), 66–75.

96. Jasmin C. Blanchette, Cezary Kaliszyk, L. C. Paulson and Josef Urban. Hammering towards QED. *J. Formalized Reasoning* **9**:1 (2016), 101–148.

97. Mohammad Abdulaziz and L. C. Paulson. An Isabelle/HOL formalisation of Green's theorem. *In*: Jasmin C. Blanchette and Stephan Merz (editors), *ITP 2016: Seventh International Conference on Interactive Theorem Proving* (Springer LNCS 9807, 2016), 3–19.

98. Wenda Li and L. C. Paulson. A formal proof of Cauchy's residue theorem. *In*: Jasmin C. Blanchette and Stephan Merz (editors), *ITP 2016: Seventh International Conference on Interactive Theorem Proving* (Springer LNCS 9807, 2016), 235–251.

99. Zongyan Huang, Matthew England, James H. Davenport and L. C. Paulson. Using machine learning to decide when to precondition cylindrical algebraic decomposition with Groebner bases. *SYNASC 2016: Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, (2016), 45–52.

100. L. C. Paulson. Computational logic: its origins and applications. *Proceedings of the Royal Society Series A* **474**:2210 (2018).

101. L. C. Paulson. Michael John Caldwell Gordon (FRS 1994), 28 February 1948 – 22 August 2017. *Biographical Memoirs of Fellows of the Royal Society* **65** (2018), 89–113.

102. Wenda Li and L. C. Paulson. Counting polynomial roots in Isabelle/HOL: a formal proof of the Budan-Fourier theorem. *In*: Assia Mahboubi and Magnus Myreen (editors), CPP 2019 — The 8th ACM SIGPLAN International Conference on Certified Programs and Proofs (2019), 52–64.

103. Wenda Li, Grant Olney Passmore and L. C. Paulson. Deciding univariate polynomial problems using untrusted certificates in Isabelle/HOL. *J. Automated Reasoning* **62** (2019), 69–91.

104. Mohammad Abdulaziz and L. C. Paulson. An Isabelle/HOL formalisation of Green's theorem. *J. Automated Reasoning* **63**:3 (2019), 763–786.

105. Zongyan Huang, Matthew England, David Wilson, James H. Davenport and L. C. Paulson. Using machine learning to improve cylindrical algebraic decomposition. *Mathematics in Computer Science* **13**:4 (2019), 461–488.

106. Lawrence C. Paulson, Tobias Nipkow and Makarius Wenzel. From LCF to Isabelle/HOL. *Formal Aspects of Computing* **31**:6 (2019), 675–698.

107. L. C. Paulson. Formalising mathematics in simple type theory. *In*: Stefania Centrone, Deborah Kant and Deniz Sarikaya (editors), *Reflections on the Foundations of Mathematics: Univalent Foundations, Set Theory and General Thoughts* (Springer, 2019), 437–453.

108. Wenda Li and L. C. Paulson. Evaluating winding numbers and counting complex roots through Cauchy indices in Isabelle/HOL. *J. Automated Reasoning* **64**:2 (2020), 331–360.

109. Paulo Emílio de Vilhena and L. C. Paulson. Algebraically closed fields in Isabelle/HOL. *In*: Nicolas Peltier and Viorica Sofronie-Stokkermans (editors), *IJCAR 2020 — Automated Reasoning, 10th International Joint Conference*, Part II (Springer LNCS 12167, 2020), 204–220.

Workshop Papers

1. L. C. Paulson. Inductive analysis of the internet protocol TLS. *In*: Bruce Christianson, Bruno Crispo, William S. Harbison and Michael Roe (editors), *Security Protocols: 6th International Workshop 1998* (Springer LNCS 1550, 1999), 1–12.

2. L. C. Paulson. Relations between secrets: the Yahalom protocol (extended abstract). *In*: Bruce Christianson, Bruno Crispo, James A. Malcolm and Michael Roe (editors), *Security Protocols: 7th International Workshop 1999* (Springer LNCS 1550, 2000), 73–77.

3. Giampaolo Bella and L. C. Paulson. Making sense of specifications: the formalization of SET. *In*: Bruce Christianson, Bruno Crispo, James A. Malcolm and Michael Roe (editors) *Security Protocols: 8th International Workshop 2000* (Springer LNCS 2133, published 2001), 74–81.

4. Giampaolo Bella and L. C. Paulson. A proof of non-repudiation. *Security Protocols: 9th International Workshop* 2001 (Springer LNCS 2467, published 2002), 119–125.

5. Sidi O. Ehmety and L. C. Paulson. Representing component states in higher-order logic. *In*: Richard J. Boulton and Paul B. Jackson (editors), *Theorem Proving in Higher Order Logics* (2001).

6. Christoph Benzmüller, L. C. Paulson, Frank Theiss and Arnaud Fietzke. Progress Report on Leo-II, an Automatic Theorem Prover for Higher-Order Logic. *In*: Klaus Schneider and Jens Brandt (editors), Theorem Proving in Higher Order Logics 2007 — Emerging Trends.

7. Jean E. Martina and L. C. Paulson. Verifying multicast-based security protocols using the inductive method. CryptoForma Workshop 2011: Workshop on Formal Methods and Cryptography, Limerick, June 2011.

8. Ralph Romanos, L. C. Paulson. Proving the impossibility of trisecting an angle and doubling the cube. Isabelle Users Workshop (associated with the conference ITP 2012). Princeton, New Jersey, August 2012.

9. Agnieszka Słowik, Chaitanya Mangla, Mateja Jamnik, Sean Holden and L. C. Paulson. Bayesian optimisation for premise selection in automated theorem proving (Student abstract). *In AAAI Conference on Artificial Intelligence* (AAAI 2020). AAAI Press. Online at `https://doi.org/10.1609/aaai.v34i10.7232` (2020).

10. L. C. Paulson. Ackermann's function in iterative form: a subtle termination proof with Isabelle/HOL. Isabelle Workshop 2020.

11. Yiannos Stathopoulos, Angeliki Koutsoukou-Argyraki and L. C. Paulson. SErAPIS: A concept-oriented search engine for the Isabelle libraries based on natural language. Isabelle Workshop 2020.

12. Chaitanya Mangla, Sean B. Holden and L. C. Paulson. Bayesian optimisation of solver parameters in CBMC. 18th International Workshop on Satisfiability Modulo Theories (SMT), 2020.

Invited and Other Unrefereed Publications

1. L. C. Paulson. Keep E-Journals Affordable. Letter, *Communications of the ACM* **44**:8 (2001), 11–13.
2. L. C. Paulson. The Descent of BAN. *In*: Andrew Herbert and Karen Spärck Jones (editors), *Computer Systems: Theory, Technology, and Applications* (Springer, 2004), 225–228.
3. L. C. Paulson. Welcome Defects as a Sign of Innovation? Letter, *Communications of the ACM* **48**:11 (2005), 11–13.
4. Markus Wenzel and L. C. Paulson. Isabelle/Isar. *In*: Freek Wiedijk (editor), *The Seventeen Provers of the World* (Springer LNCS 3600, 2006), 41–49.
5. L. C. Paulson. Even a Good Abstraction Needs Experience and Testing, Too. Letter, *Communications of the ACM* **50**:9 (2007), 13–14.
6. Makarius Wenzel, L. C. Paulson, Tobias Nipkow. The Isabelle Framework. *In*: Otmane Ait Mohamed, César Muñoz, Sofiène Tahar (editors), *Theorem Proving in Higher Order Logics* (Springer LNCS 5170, 2008), 33–38.
7. Christoph Benzmüller and L. C. Paulson. Exploring Properties of Normal Multimodal Logics in Simple Type Theory with LEO-II. In: Chris Benzmüller, Chad Brown, Jörg Siekmann and Rick Statman (editors), *Reasoning in Simple Type Theory: Festschrift in Honor of Peter B. Andrews on His 70th Birthday* (College Publications, 2008), 401–422.
8. Christoph Benzmüller, L. C. Paulson. Quantified Multimodal Logics in Simple Type Theory. CoRR abs/0905.2435. `http://arxiv.org/abs/0905.2435` (2009). Subsequently accepted to Logica Universalis (2012).
9. L. C. Paulson. Functional Programming in ML. In Phillip Laplante (editor), *Encyclopedia of Software Engineering* (Taylor & Francis, 2010), 333–346.
10. L. C. Paulson. What Liability for Faulty Software? Letter, *Communications of the ACM* **55**:2 (2012), 6–7.
11. L. C. Paulson. What We Want from Computer Science. Letter, *Communications of the ACM* **56**:7 (2013), 8–9.
12. L. C. Paulson. Provenance of British Computing. Letter, *Communications of the ACM* **57**:9 (2014), 8–9.
13. L. C. Paulson. Abolish Software Warranty Disclaimers. Letter, *Communications of the ACM* **58**:5 (2015), 8–9.
14. L. C. Paulson. Time to Retire 'Computational Thinking'? Letter, *Communications of the ACM* **60**:9 (September 2017), 8–9.
15. L. C. Paulson. Predicting Failure of the University. Letter, *Communications of the ACM* **61**:4 (April 2018), 8–9.

Technical Reports

Reports subsequently published elsewhere are omitted.

1. L. C. Paulson. Recent developments in LCF: examples of structural induction. Report 34, Computer Lab (1983).
2. L. C. Paulson. Rewriting in Cambridge LCF. Report 35, Computer Lab (1983).
3. L. C. Paulson. The revised logic PPLAMBDA. Report 36, Computer Lab (1983).
4. L. C. Paulson. Tactics and tacticals in Cambridge LCF. Report 39, Computer Lab (1983).
5. L. C. Paulson. Interactive theorem proving with Cambridge LCF. Report 80, Computer Lab (1985).
6. G. Cousineau, M. Gordon, G. Huet, R. Milner, L. C. Paulson, C. Wadsworth. The ML Handbook. Technical Report, INRIA-Rocquencourt, May 1985.
7. L. C. Paulson. The representation of logics in higher-order logic. Report 113, Computer Lab (1987).
8. L. C. Paulson. A preliminary user's manual for Isabelle. Report 133, Computer Lab (1988).
9. L. C. Paulson and Tobias Nipkow. Isabelle tutorial and user's manual. Report 189, Computer Lab (1990).
10. L. C. Paulson. Set theory as a computational logic: I. From foundations to functions. Report 271, Computer Lab (1992).
11. L. C. Paulson. Introduction to Isabelle. Report 280, 2nd edition, Computer Lab (1993).
12. L. C. Paulson. Isabelle reference manual. Report 283, 2nd edition, Computer Lab (1993).
13. L. C. Paulson. Isabelle's object-logics. Report 286, 2nd edition, Computer Lab (1993).
14. L. C. Paulson. A selection from the proof development libraries of Isabelle. Report, Computer Lab (1993).
15. L. C. Paulson (editor). Proceedings of the First Isabelle Users Workshop. Report 379, Computer Lab (1995).
16. L. C. Paulson. Mechanized proofs of security protocols: Needham-Schroeder with public Keys. Report 413, Computer Lab (1997).
17. Giampaolo Bella, Fabio Massacci and L. C. Paulson. Verifying the SET Purchase Protocols. Report 524, Computer Lab (2001).

Book Reviews

1. *Logical Frameworks* by G. Huet & G. Plotkin, *Computer J.* **35** (1992), 156.
2. *Functional Programming with Hope* by R. Bailey, *Computer J.* **35** (1992), 491.
3. *Type Theory and Functional Programming* by S. Thompson and *Constructive Foundations for Functional Languages* by R. Turner, *Formal Aspects of Computer Science* **4** (1992), 495–6.
4. *The Little MLer* by M. Felleisen and D. P. Friedman, *Computer J.* **41** (1998), 425.

Printed on March 4, 2021