Matt Kaufmann

Senior Research Scientist, Retired
Dept. of Computer Science, Univ. of Texas at Austin

2203 Euclid Avenue, Austin, TX 78704
(512) 443-9212
Born: 12/9/52; married; no children
U.S. Citizen

Education

Ph.D., Mathematics, June, 1978
University of Wisconsin, Madison, Wisconsin

M.A., Mathematics, December, 1974
University of Wisconsin, Madison, Wisconsin

S.B., Mathematics, June, 1973
Massachusetts Institute of Technology

Honors

Co-winner (with Robert S. Boyer and J Strother Moore) of the 2005 ACM Software System Award.

Support Activities

Have served on program committees, Ph.D. committees, and NSF panels; reviewed papers; served as conference/workshop (co-)chair; maintained mailing lists; etc. Details available upon request.

Employment History

Senior Research Scientist, December 2005 – May 2019
Dept. of Computer Sciences, University of Texas, Austin, Texas

Senior Member of the Technical Staff, August 1999 – November 2005
Advanced Micro Devices, Inc., Austin, Texas

Senior Systems Engineer, August 1997 – August 1999
EDS, Inc., CIO Services, Austin, Texas

Senior Individual Contributor, August 1995 – August 1997
Motorola, Inc., Austin, Texas

Senior Computing Research Scientist, September 1987 – August 1995
Computational Logic, Inc., Austin, Texas

Research Scientist, August 1986 – August 1987
Institute for Computing Science, University of Texas, Austin, Texas

Adjunct Associate Professor: Spring 1988, Fall 1989, and Fall 1994
Departments of Mathematics (1988, 89) and Philosophy (1994)
University of Texas, Austin, Texas

Research Scientist, October 1985 – June 1986
Associate Research Scientist, June 1984 – October 1985
Austin Research Center
Burroughs Corporation, Austin, Texas

Assistant Professor, August 1978 – May 1984 (on leave 9/82 – 5/83)
Department of Mathematics
Purdue University, West Lafayette, Indiana

Visiting Assistant Professor, September 1982 – May 1983
Department of Mathematics
University of Connecticut, Storrs, Connecticut

Teaching Assistant, August 1973 – May 1978
Department of Mathematics
University of Wisconsin, Madison, Wisconsin

Summary of Experience

Tool Development:

Co-author (with J Moore) of the ACL2 theorem proving system, a successor to the so-called Boyer-Moore theorem prover. Its source code contains more than 250,000 lines (11+ megabytes) as of Version 8.3, not including another 130,000 lines of documentation. Also have been contributing actively to the ACL2 community through the ACL2 Workshop, the acl2-help mailing list, the Univ. of Texas ACL2 seminar, the ACL2 Slack channel, and serving on Ph.D. committees.

Other formal verification tools

- Wrote many tools based on ACL2: some contributions have been relatively recent, for example, including a program transformation tool based on simplification and an efficient, formally verified SAT proof-checker; while others were made decades ago, for example, a symbolic execution tool for ACL2 for use in CLI/Motorola DSP verification project (early to mid 1990s).
- Enhanced Motorola's Verilog/DSL model-checker and compiler, substantially improving efficiency, robustness, features, output, and documentation
- Built numerous extensions to the Boyer-Moore "Nqthm" prover, including an interactive enhancement, a quantification capability, and many other tools
- Designed and implemented a prototype verification system for a subset of Common Lisp that includes macros and imperative features such as assignment, global variables, iterative control, and property lists.
- With Bob Boyer, developed a modification of the Boyer-Moore theorem prover to use as a verification tool for the applicative language SASL. Worked out theoretical details proving soundness of the approach.

Translators

- At AMD, co-developed a language and tools for representing state-machine HDL descriptions in ACL2
- Wrote internal Motorola translator from one commercial hardware design language to another
- Wrote translators for several small hardware design languages (Russinoff's to VHDL, MIMIC to Brock-Hunt, Brock-Hunt to VHDL).
- Wrote translator from Nqthm to ACL2
- Co-designed translation of Gypsy programming language into Nqthm

Other tools

- At AMD, wrote sophisticated C++ multi-processor memory model and did associated simulation debug. This tool is used in both standalone runs (supporting the golden x86 model) and simulation runs. Wrote thorough documentation, including supporting theory with proofs.
- Became AMD expert on routing tables and developed several pertinent tools, including a legality checker
- Wrote heuristics-based tool at AMD for analyzing certain collected Northbridge transaction data, gave assistance in its use, and responded to requests for analysis of its output
- At AMD, developed and ran a "smart" tool that finds classes of syntax bugs in rtl

- Wrote tool for finding dead code in RTL
- Developed numerous enhancements of EDS analysis tool for COBOL, Cogen 2000^{TM}. In particular, implemented data flow, type propagation, and test instrumentation. Also developed regression test capability.
- Wrote assembler for Motorola CAP processor (assisted by Bishop Brock) in ACL2
- Wrote symbolic reducer for a variant of the Micro-Gypsy language
- Co-authored a course-grained parallel "dispatcher" and an application of that system to provide course-grained parallelism in the Boyer-Moore theorem prover
- Developed and documented an enhanced Lisp tracing facility, now part of the Gnu Common Lisp distribution, together with a tracing facility for execution of the Boyer-Moore logic
- Implemented the PODEM algorithm for generating tests for faults in circuits
- Wrote a symbolic reducer for a fragment of the applicative language SASL, in SASL

Applications of automated reasoning tools:

Formally verified the correctness of a SAT proof-checker, used for example in SAT Competition 2017 and in verifying "World's Largest Math Proof" (194 terabytes, 8,651 CPU hours).

Contributed to x86 ISA modeling project at UT Austin.

At AMD, using ACL2:

Did some proofs about floating-point rtl, and improved ACL2 rtl library in the process.

Did some proof work for a bus unit's rtl

Wrote high-level protocol spec in ACL2-based modeling language and did some proof work on it, developing about 10 invariants

Completed a write-ordering proof

Using ACL2, specified and verified expression replacement rules used by EDS tool Cogen 2000 for Year 2000 remediation

Created, with J Moore, a mechanically-checked proof of correctness of the floating-point division algorithm, as described by its designer, for the microcode of the AMD K5 microprocessor

Co-authored a simplified 60x (Power PC) bus protocol specification and verified properties of it using a model-checker

Improved ACL2 integer libraries

Contributed to FM9001 chip verification

Assisted in Piton assembler proof

Developed and published correctness proof for a generalization algorithm

Co-developed Ada subset semantics and did corresponding program verification

Verified SASL (lazy functional language) programs

Did preliminary non-standard analysis verification work

Instructional, support, and tech transfer activities:

At AMD, in collaboration with others, developed theory and ran tools pertaining to routing of packets, including x86 assembly and writing descriptions for BIOS guide. (See also related item about routing, above, under "Tool Development / Other Tools.")

Made extensive comments on AMD protocol documentation

Maintained rules documents for Year 2000 COBOL renovations performed by EDS CIO Services

Assisted over two dozen internal Motorola customers in use of formal verification and translator tools

Developed tutorial materials for the Boyer-Moore theorem prover

By invitation, visited the Mathematics Reasoning group IRST (Trento, Italy) for 6 days in July, 1994 and for several days in June, 1991

Gave short courses in Lisp and theorem proving for hardware verification at Boeing Corp. and at Motorola GSG (Phoenix)

Assisted in Hardware Verification Institute at UT Year of Programming, 1987

Gave several lectures for an in-house course on the programming language SASL

Taught 3 logic courses at University of Texas (2 graduate, 1 undergrad)

While on the Mathematics faculty at Purdue University: taught numerous mathematics courses at many levels, supervised teaching assistants, served on departmental textbook selection committees, and reviewed college mathematics textbooks.

As a teaching assistant in Mathematics at the University of Wisconsin, taught algebra and trigonometry courses, taught discussion sections of calculus courses, and coordinated College Algebra course

Other technical activities:

At AMD, implemented C++ northbridge-related checkers and did associated simulation debug

Participated in evaluations of formal and semi-formal tools external to AMD

Wrote scripts in support of Motorola's model checker, for example to support regression testing and tool release

Participated in comparisons of theorem provers

Investigated hardware description language VHDL from standpoint of formal reasoning, and (with Bill Young) documented findings

Evaluated TRW's "Deductive Theory Manager" and documented findings

Developed a theoretical basis for well-formedness in the Gypsy prover

Served by invitation on NSF Committee of Visitors, July, 1996. Evaluated NSF programs in the Division of Computer and Computation Research (Numeric, Symbolic, and Geometric Computation).

Served on committees (program, dissertation, ...), refereed papers, did proposal writing, gave invited talks

Developed semantics for "Nqthm" version of the Boyer-Moore logic, including a proof of soundness, and contributed to work on *functional instantiation* in Nqthm

Assisted in code review of Nqthm prover

Developed a formal logic and model theory for the programming language SASL

Co-developed improved compilation algorithm for SASL that eliminates unsoundness in an earlier algorithm, with virtually negligible loss of efficiency

Ran experiments to analyze potential speedup for concurrent execution of SASL programs

Pure mathematics research included over 20 papers (some co-authored) in Mathematical Logic (see list below)

Invited Talks

Logical Foundations for the ACL2 Theorem Prover. Invited talk, JAF ("Weak Arithmetics Days"), May 28, 2019, New York City. URL https://www.cs.utexas.edu/users/kaufmann/talks/acl2-jaf-2019-nyc/index.html.

ACL2 Support for Automated and Interactive Proof. Invited talk, 14th KeY Symposium 2015, Gothenburg, Sweden, July 27, 2015. URL http://www.cs.utexas.edu/users/kaufmann/talks/key-invited-2015/index.html.

Implementation of a Computational Logic. Invited talk, Logic Seminar, Univ. of Gothenburg, Gothenburg, Sweden, June, 2015). URL http://www.cs.utexas.edu/users/kaufmann/talks/acl2-for-logicians/acl2-comp-logic.pdf.

Verifying LabVIEW Graphical Programs with ACL2, invited talk given at *Workshop on Linking Tools for Verified Software*. Jim Woodcock, Mike Gordon, Tony Hoare, organizers; November 27-28, 2008, Cambridge, UK. URL http://www.cl.cam.ac.uk/∼mjcg/GC6/Meeting.27-28.11.08.html.

Aspects of ACL2 User Interaction. Invited Talk, 8th International Workshop On User Interfaces for Theorem Provers (UITP 2008), Montreal, Canada, August, 2008. URL http://www.ags.uni-sb.de/%7Eomega/workshops/UITP08/kaufmann-UITP08/talk.html.

An ACL2 Tutorial (with J Strother Moore). Invited talk; Proceedings of Theorem Proving in Higher Order Logics, 21st International Conference, TPHOLs 2008, Montreal, Canada, August, 2008, pp. 17–21. URL http://dx.doi.org/10.1007/978-3-540-71067-7_4.

Automated Reasoning and The ACL2 Theorem Proving System (with J Strother Moore). The 2006 Visions of Computing Lecture Series, Department of Computer Sciences, University of Texas at Austin, November, 2006. URL http://www.cs.utexas.edu/users/kaufmann/visions-moore-kaufmann.pdf.

Maintaining the ACL2 Theorem Proving System (with J Strother Moore). Invited talk. *Proceedings of the FLoC'06 Workshop on Empirically Successful Computerized Reasoning, 3rd International Joint Conference on Automated Reasoning* (G. Sutcliffe, R. Schmidt, and S. Schulz, editors), CEUR Workshop Proceedings Vol. 192, pp. 1-17, Seattle, Washington, August 2006. http://CEUR-WS.org/Vol-192/.

ACL2 Support for Verification Projects. Invited talk, *Proc. 15th Intl. Conf. on Automated Deduction*, ed. C. Kirchner and H. Kirchner, Lec. Notes Artif. Intelligence 1421, Springer-Verlag, Berlin, 1998, pp. 220–238.

An Informal Discussion of Issues in Mechanically-assisted Reasoning. Keynote Address. In: *Proceedings of the 1991 International Workshop on the HOL Theorem Proving System and its Applications*. URL http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=596297&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D596297

Publications

NOTE: *Immediately below is a list of publications, including books as well as articles submitted to or published in journals and proceedings. After this list appear three lists of technical reports, which do not include approximately 50 internal notes at Computational Logic, Inc., nor several Motorola internal documents.*

Extended Abstract: Stobj-tables (with Rob Sumners and Sol Swords). In: *Proceedings of the 17th International Workshop on the ACL2 Theorem Prover and Its Applications*, May 26-27, 2022. *Electronic Proceedings in Theoretical Computer Science.*, Rob Sumners and Cuong Chau, editors. Vol. 359, 2022.
URL `https://cgi.cse.unsw.edu.au/~eptcs/content.cgi?ACL22022#EPTCS359.1`.

Extended Abstract: Iteration in ACL2, WITH .. DO (with J Strother Moore). In: *Proceedings of the 17th International Workshop on the ACL2 Theorem Prover and Its Applications*, May 26-27, 2022. *Electronic Proceedings in Theoretical Computer Science.*, Rob Sumners and Cuong Chau, editors. Vol. 359, 2022.
URL `https://cgi.cse.unsw.edu.au/~eptcs/content.cgi?ACL22022#EPTCS359.2`.

Iteration in ACL2 (with J Strother Moore). In: *Proceedings of the 16th International Workshop on the ACL2 Theorem Prover and Its Applications*, May 28-29, 2020. *Electronic Proceedings in Theoretical Computer Science.*, Ruben Gamboa and Grant Passmore, editors. Vol. 327, pp. 16-31, 2020.
URL `http://eptcs.web.cse.unsw.edu.au/paper.cgi?ACL22020.2.pdf`.

Limited Second-Order Functionality in a First-Order Setting (with J Strother Moore). *Journal of Automated Reasoning*, Springer, December 2018, pp. 1–32.
URL `https://doi.org/10.1007/s10817-018-09505-9`.

DefunT: A Tool for Automating Termination Proofs by Using the Community Books (Extended Abstract). In: *Proceedings of the 15th International Workshop on the ACL2 Theorem Prover and Its Applications*, Austin, Texas, USA, November 5-6, 2018. *Electronic Proceedings in Theoretical Computer Science.*, Matt Kaufmann and Shilpi Goel, editors. Vol. 280, pp. 161-163, 2018.
URL `https://cgi.cse.unsw.edu.au/~eptcs/paper.cgi?ACL22018.12`.

Data-Loop-Free Self-Timed Circuit Verification (with Cuong Chau, Warren A. Hunt, Jr., Marly Roncken, and Ivan Sutherland). ASYNC 2018, Vienna, Austria, 2018.

Efficient, Verified Checking of Propositional Proofs (with Marijn Heule, Warren Hunt, Jr., and Nathan Wetzler). Interactive Theorem Proving – ITP 2017. LNCS 10499, pp. 269-284, Springer International Publishing, 2017.

Efficient Certified RAT Verification (with Luís Cruz-Filipe, Marijn Heule, Warren Hunt, and Peter Schneider-Kamp). Proceedings CADE 26 - 26th International Conference on Automated Deduction, Gothenburg, Sweden, August 6-11, 2017, Leonardo de Moura, editor,

pp. 220–236.
URL https://doi.org/10.1007/978-3-319-63046-5_14.

A Versatile, Sound Tool for Simplifying Definitions (with Alessandro Coglio and Eric W. Smith). ACL2 Workshop 2017.

Meta-extract: Using Existing Facts in Meta-reasoning (with Sol Swords). ACL2 Workshop 2017.

Iterated Ultrapowers for the Masses (with Ali Enayat and Zachiri McKenzie). *Archive for Mathematical Logic*, Springer.
URL http://link.springer.com/article/10.1007/s00153-017-0592-1 (Springer Open Access).
Preliminary version: URL http://arxiv.org/abs/1702.03487.

Largest Initial Segments Pointwise Fixed by Automorphisms of Models of Set Theory (with Ali Enayat and Zachiri McKenzie). *Archive for Mathematical Logic*, Springer.
URL http://link.springer.com/article/10.1007/s00153-017-0582-3
(Springer Open Access). Preliminary version: URL https://arxiv.org/abs/1606.04002.

Industrial Hardware and Software Verification with ACL2 (with Warren A. Hunt, Jr., J Strother Moore, and Anna Slobodova). In: *Verified Trustworthy Software Systems* (Gardner, P., O'Hearn, P., Gordon, M., Morrisett, G. and Schneider, F.B., Eds), *Philosophical Transactions A*, vol 374, Royal Society Publishing, DOI 10.1098/rsta.2015.0399, September, 2017.
URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5597723/pdf/rsta20150399.pdf.

Engineering a Formal, Executable x86 ISA Simulator for Software Verification (with Shilpi Goel and Warren A. Hunt, Jr.). *Provably Correct Systems* (ed. Mike Hinchey, Jonathan P. Bowen, and and Ernst-Rüdiger Olderog), 2017, Springer, 173–209.

Well-Formedness Guarantees for ACL2 Metafunctions and Clause Processors (with J Strother Moore). DIFTS'15. URL http://www.faculty.ece.vt.edu/chaowang/difts2015/papers/paper_1.pdf.

Fourier Series Formalization in ACL2(r) (with Cuong K. Chau and Warren A. Hunt, Jr.). *Proceedings of ACL2 Workshop 2015*, Matt Kaufmann and David L. Rager, editors. *Electronic Proceedings in Theoretical Computer Science.*, Vol. 192, pp. 35-51, 2015.
URL http://dx.doi.org/10.4204/EPTCS.192.4.

Rough Diamond: An Extension of Equivalence-based Rewriting (with J Strother Moore). *Proceedings of ITP 2014, 5th Conference on Interactive Theorem Proving,* Gerwin Klein and Ruben Gamboa, editors. LNCS 8558 pp. 537-542, Springer International Publishing, 2014. DOI 10.1007/978-3-319-08970-6_35. URL http://dx.doi.org/10.1007/978-3-319-08970-6_35.

Industrial-Strength Documentation for ACL2 (with Jared Davis). *Proceedings of ACL2*

Workshop 2014, Julien Schmaltz and Freek Verbeek, editors. *Electronic Proceedings in Theoretical Computer Science,* pp. 9-25. DOI 10.4204/EPTCS.152.2. URL http://arxiv.org/abs/1406.2266.

Enhancements to ACL2 in Versions 6.2, 6.3, and 6.4 (with J Strother Moore). *Proceedings of ACL2 Workshop 2014,* Julien Schmaltz and Freek Verbeek, editors. *Electronic Proceedings in Theoretical Computer Science,* pp. 1-7. DOI 10.4204/EPTCS.152.1. URL http://arxiv.org/abs/1406.1556v1.

Simulation and Formal Verification of x86 Machine-Code Programs that make System Calls (with Shilpi Goel, Warren A. Hunt, Jr., and Soumava Ghosh). *Proceedings of Formal Methods in Computer-Aided Design (FMCAD'14)*, October, 2014. URL http://www.cs.utexas.edu/users/hunt/FMCAD/FMCAD14/proceedings/18_goel.pdf.

A Parallelized Theorem Prover for a Logic with Parallel Execution (with David L. Rager and Warren A. Hunt, Jr.). *Proceedings of ITP 2013, 4th Conference on Interactive Theorem Proving.* S. Blazy, C. Paulin-Mohring, and D. Pichardie (Eds.), LNCS 7998, pp. 435-450, Springer-Verlag Berlin Heidelberg 2013.

Abstract Stobjs and Their Application to ISA Modeling (with Shilpi Goel and Warren A. Hunt, Jr.). In: Proceedings of ACL2 Workshop 2013, Ruben Gamboa and Jared Davis, editors. *Electronic Proceedings in Theoretical Computer Science*, Volume 114, pp. 54-69. DOI 10.4204/EPTCS.114.5. URL http://eptcs.org/content.cgi?ACL22013.

Enhancements to ACL2 in Versions 5.0, 6.0, and 6.1 (with J Strother Moore). In: Proceedings of ACL2 Workshop 2013, Ruben Gamboa and Jared Davis, editors. *Electronic Proceedings in Theoretical Computer Science*, Volume 114, pp. 5-12. DOI 10.4204/EPTCS.114.1. URL http://eptcs.org/content.cgi?ACL22013.

A Formal Model of a Large Memory that Supports Efficient Execution (with Warren A. Hunt, Jr.). *Proceedings of Formal Methods in Computer-Aided Design (FMCAD'12)* (G. Cabodi and S. Singh, editors). ACM Digital Library.
URL http://www.cs.utexas.edu/users/hunt/FMCAD/FMCAD12/fmcad2012.pdf, pp. 60-67, 2012.

How Can I Do That with ACL2? Recent Enhancements to ACL2 (with J Strother Moore). In Proceedings 10th International Workshop on the ACL2 Theorem Prover and its Applications, Austin, Texas, USA, November 3-4, 2011, David Hardin and Julien Schmaltz, editors. *Electronic Proceedings in Theoretical Computer Science*, Volume 70, pp. 46-60. DOI 10.4204/EPTCS.70.1.
URL http://eptcs.org/content.cgi?ACL22011.

Integrating Testing and Interactive Theorem Proving (with Harsh Raju Chamarthi, Peter C. Dillinger, and Panagiotis Manolios). In Proceedings 10th International Workshop on the ACL2 Theorem Prover and its Applications, Austin, Texas, USA, November 3-4, 2011. Proceedings of ACL2 Workshop 2011, David Hardin and Julien Schmaltz, editors. *Electronic*

Proceedings in Theoretical Computer Science, Volume 70, pp. 4-19. DOI 10.4204/EPTCS.70.1.
URL http://eptcs.org/content.cgi?ACL22011.

A Futures Library and Parallelism Abstractions for a Functional Subset of Lisp (with David L. Rager and Warren A. Hunt, Jr.). In proceedings of ELS 2011 (4th European Lisp Symposium), March 31 – April 1, 2011, Hamburg, Germany, pp. 13-16.
URL http://www.european-lisp-symposium.org/editions/2011/ELS2011.pdf.

Interactive Theorem Proving: First International Conference, ITP 2010, Edinburgh, Scotland, July 2010 (co-editor with Lawrence Paulson). LNCS 6172, Springer, 2010 (eBook at URL http://dx.doi.org/10.1007/978-3-642-14052-5).

The Right Tools for the Job: Correctness of Cone of Influence Reduction Proved Using ACL2 and HOL4 (with M. Gordon and S. Ray). *Journal of Automated Reasoning*, Volume 47, Number 1, Springer, 2011, pp. 1–16, DOI 10.1007/s10817-010-9169-y.

ACL2 and Its Applications to Digital System Verification (with J Strother Moore). In: *Design and Verification of Microprocessor Systems for High-Assurance Applications*, David S. Hardin, ed., Springer, 2010, pp. 1–21.

Abbreviated Output for Input in ACL2: An Implementation Case Study. In Proceedings of ACL2 Workshop 2009, URL https://www.cs.utexas.edu/users/moore/acl2/workshop-2009/final/10/10.pdf.

Formal Verification of LabVIEW Programs Using the ACL2 Theorem Prover (with Jacob Kornerup and Mark Reitblatt). In Proceedings of ACL2 Workshop 2009, URL http://www.cs.utexas.edu/users/sandip/acl2-09/.

Formalizing Routing Models in ACL2 (with Warren A. Hunt, Jr., Robert Bellarmine Krug, and Sandip Ray). Technical Report TR-08-11, Department of Computer Sciences, University of Texas at Austin, March 2008.
URL ftp://ftp.cs.utexas.edu/pub/techreports/tr08-11.pdf.

Proof Search Debugging Tools in ACL2 (with J Moore). *Tools and Techniques for Verification of System Infrastructure, A Festschrift in honour of Prof. Michael J. C. Gordon FRS* (Richard Boulton, Joe Hurd, and Konrad Slind, organizers). March 25-26, 2008, Royal Society, London.

Hacking and Extending ACL2 (with Peter Dillinger and Panagiotis Manolios). In Proceedings of ACL2 Workshop 2007, URL
http://www.cs.uwyo.edu/∼ruben/acl2-07/Main/AdvanceProgram.

Proof Pearl: Wellfounded Induction on the Ordinals up to ε_0 (with Konrad Slind). Proceedings of Theorem Proving in Higher Order Logics, 20th International Conference, TPHOLs 2007, Kaiserslautern, Germany, September 10-13, 2007. LNCS 4732, Springer, pp. 294–301.

Integrating External Deduction Tools with ACL2 (with J S. Moore, Sandip Ray, and Erik

Reeber). *Journal of Applied Logic* (Special Issue: Empirically Successful Computerized Reasoning), Volume 7, Issue 1, March 2009, pp. 3–25. Also published online (DOI 10.1016/j.jal.2007.07.002). Preliminary version in: *Proceedings of the 6th International Workshop on the Implementation of Logics (IWIL 2006)* (C. Benzmüller, B. Fischer, and G. Sutcliffe, editors), CEUR Workshop Proceedings Vol. 212, Phnom Penh, Cambodia, pp. 7-26, November 2006. http://ceur-ws.org/Vol-212/.

An Integration of HOL and ACL2 (with Michael J.C. Gordon, Warren A. Hunt, Jr., and James Reynolds). *Proceedings of Formal Methods in Computer-Aided Design (FMCAD'06)* (A. Gupta and P. Manolios, editors). IEEE Computer Society Press, pp. 153-160, November, 2006.

Integrating CCG analysis into ACL2 (with Panagiotis Manolios, J Moore, and Daron Vroon). *Proceedings of The Eighth International Workshop on Termination*, pp. 64-68, August, 2006.

An Embedding of the ACL2 Logic in HOL (with Michael J.C. Gordon, Warren A. Hunt, Jr., and James Reynolds). *Proceedings of ACL2 Workshop 2006*, August, 2006. ACM Digital Library,
URL http://portal.acm.org/toc.cfm?id=1217975.

Double Rewriting for Equivalential Reasoning in ACL2 (with J Strother Moore). *Proceedings of ACL2 Workshop 2006*, August, 2006. ACM Digital Library,
URL http://portal.acm.org/toc.cfm?id=1217975.

Efficient Execution in an Automated Reasoning Environment (with David A. Greve, Panagiotis Manolios, J Strother Moore, Sandip Ray, José Luis Ruiz-Reina, Rob Sumners, Daron Vroon, and Matthew Wilding). Journal of Functional Programming, Volume 18, Issue 01, January 2008, Cambridge University Press. Long version is available as Technical Report TR-06-59, Department of Computer Sciences, University of Texas at Austin,
URL ftp://ftp.cs.utexas.edu/pub/techreports/tr06-59.pdf.

Rewriting with Equivalence Relations in ACL2 (with Bishop Brock and J Strother Moore). *Journal of Automated Reasoning* 40 (2008), pp. 293-306. Also published online (DOI 10.1007/s10817-007-9095-9).

Meta Reasoning in ACL2 (with Warren Hunt, Robert Krug, J Moore and Eric Smith). TPHOLs 2005, ed. J. Hurd and T. F. Melham, LNCS 3603, Springer-Verlag, Berlin, 2005, pp. 163-178.

Formal Verification of Floating-Point RTL at AMD using the ACL2 Theorem Prover (David Russinoff, Matt Kaufmann, Eric Smith, Robert Sumners). 17th IMACS World Congress: Scientific Computation, Applied Mathematics and Simulation. July, 2005. Available from URL http://www.russinoff.com/papers/paris.pdf.

Some Key Research Problems in Automated Theorem Proving for Hardware and Software Verification (with J Strother Moore). *Revista de la Real Academia de Ciencias (RACSAM)*,

Vol. 98, No. 1, pp. 181–196, 2004. Spanish Royal Academy of Science. Available from URL http://www.rac.es/ficheros/doc/00156.pdf.

A Tool for Simplifying Files of ACL2 Definitions. In Proceedings of ACL2 Workshop 2003, URL http://www.cs.utexas.edu/users/moore/acl2/workshop-2003/contrib/kaufmann/paper.pdf.

A Computational Logic for Applicative Common Lisp (with J Moore). In: A Companion to Philosophical Logic, D. Jacquette (ed), Blackwell Publishers, pp. 724-741, 2002.

Adding a Total Order to ACL2 (With P. Manolios). In Proceedings of ACL2 Workshop 2002, URL http://www.cs.utexas.edu/users/moore/acl2/workshop-2002/contrib/manolios-kaufmann/total-order.pdf.

Efficient Rewriting of Data Structures in ACL2 (With R. Sumners). In Proceedings of ACL2 Workshop 2002,
URL http://www.cs.utexas.edu/users/moore/acl2/workshop-2002/contrib/kaufmann-sumners/rcd.pdf.

Formal Verification of Microprocessors at AMD (with Arthur Flatau, David F. Reed, David Russinoff, Eric Smith, and Rob Sumners). Proceedings of Designing Correct Circuits 2002. URL http://www.cse.chalmers.se/∼ms/DCC02/Slides/Kaufmann.pdf.

Proceedings ACL2 Workshop 2000, Oct. 2000. Edited with J Moore. Available from URL http://apps.cs.utexas.edu/tech_reports/ncstrl/ncstrl2html.php?what=TR%20Abstracts&when=2000#UTEXAS.CS//CS-TR-00-29.

Verification of Pipeline Circuits (with David M. Russinoff). Proceedings ACL2 Workshop 2000, Oct. 2000. Available at URL http://www.cs.utexas.edu/users/moore/acl2/workshop-2000/final/russinoff-kaufmann/paper.pdf.

Computer-Aided Reasoning: An Approach (with P. Manolios and J Moore). Kluwer Academic Publishers, June, 2000.

Computer-Aided Reasoning: ACL2 Case Studies (editor, and contributed an article; with co-editors P. Manolios and J Moore). Kluwer Academic Publishers, June, 2000.

A Precise Description of the ACL2 Logic (with J Moore). Available at
URL http://www.cs.utexas.edu/users/moore/publications/km97a.ps.Z.

Structured Theory Development for a Mechanized Logic (with J Moore). *Journal of Automated Reasoning* 26, no. 2 (2001) 161-203.

Nonstandard Analysis in ACL2 (with Ruben Gamboa). *Journal of Automated Reasoning* 27(4), 323-351, 2001.

Verification of Year 2000 Conversion Rules Using the ACL2 Theorem Prover. *Software Tools for Technology Transfer* 3, no. 1 (September 2000), 13-19.

Design Constraints In Symbolic Model Checking (with Andrew Martin and Carl Pixley). In: *Computer Aided Verification: 10th International Conference* (proceedings, CAV'98 Vancouver, BC, Canada, June 28 - July 2, 1998). ed. Alan J. Hu and Moshe Y. Vardi, LNCS 1427, Springer-Verlag, 1998.

Intertwined Development and Formal Verification of a 60x Bus Model (with Carl Pixley). ICCD'97. pp. 25-30, October, 1997.

An Industrial Strength Theorem Prover for a Logic Based on Common Lisp (with J Moore). *IEEE Transactions on Software Engineering 23*, no. 4, April 1997, 203–213. (Supersedes: (with J Moore) ACL2: An Industrial Strength Version of Nqthm, In *Proceedings of the Eleventh Annual Conference on Computer Assurance (COMPASS-96)*, IEEE Computer Society Press, June, 1996, 23–34.)

Formal Verification of FIRE: A Case Study (with Jae-Young Jang, Carl Pixley, and Shaz Qadeer). In: *Proceedings of Design Automation Conference (DAC)*, 1997.

Commercial Design Verification: Methodology and Tools (with Carl Pixley, Noel R. Strader, W. C. Bruce, Jaehong Park, Kurt Shultz, Michael Burns, Jai Kumar, Jun Yuan, and Janet Nguyen). International Test Conference 1996 Proceedings, IEEE, October, 1996.

ACL2 Theorems about Commercial Microprocessors (with B. Brock and J Moore). In *Proceedings of Formal Methods in Computer-Aided Design (FMCAD'96)*, Springer-Verlag, 1996, M. Srivas and A. Camilleri (eds.), November, 1996, 275–293.

A Mechanically Checked Proof of the AMD5_K86^{TM} Floating-Point Division Program (with J Moore and T. Lynch). *IEEE Trans. Computers* 47, no. 9 (1998), pp. 913–926.

Interaction with the Boyer-Moore Theorem Prover: A Tutorial Study Using the Arithmetic-Geometric Mean Theorem (with Paolo Pecchiari). *Journal of Automated Reasoning* 16, no. 1-2 (1996) 181-222.

Design Goals for ACL2 (with J Strother Moore), in proceedings of: *Third International School and Symposium on Formal Techniques in Real Time and Fault Tolerant Systems*, Kiel, Germany (1994), pp. 92-117. Published by Christian-Albrechts-Universitat.

The Boyer-Moore Theorem Prover and Its Interactive Enhancement (with Robert S. Boyer and J Strother Moore), *Computers and Mathematics with Applications,* Vol. 29, No. 2, pp. 27-62, 1995.

An Extension of the Boyer-Moore Theorem Prover to Support First-Order Quantification, *Journal of Automated Reasoning*, Vol. 9, No. 3., December 1992, pp. 355-372.

The Role of Automated Reasoning in Integrated System Verification Environments (with Donald I. Good and J Moore), *Proceedings of TTCP XTP-1 Workshop on Effective Use of Automated Reasoning Technology in System Development*, Naval Research Laboratory, Washington DC, April 6-7, 1992.

Should We Begin a Standardization Process for Interface Logics? (with J Moore), *Proceedings of TTCP XTP-1 Workshop on Effective Use of Automated Reasoning Technology in System Development*, Naval Research Laboratory, Washington DC, April 6-7, 1992.

Functional Instantiation in First Order Logic (with Robert S. Boyer, David M. Goldschlag, and J Strother Moore), in: *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy,* Academic Press, 1991, pp. 7 - 26.

Generalization in the Presence of Free Variables: a Mechanically-Checked Correctness Proof for One Algorithm. *Journal of Automated Reasoning* 7(1991), pp. 109-158.

The Boyer-Moore Prover and Nuprl: An Experimental Comparison (with David Basin). In: *Proceedings of the First Workshop on "Logical Frameworks"*, Antibes, France, May 1990.

Demo of the Boyer-Moore Theorem Prover and some of its extensions. In: Proceedings of the First Workshop on "Logical Frameworks" (informal proceedings for Esprit Basic Research Action workshop), Antibes, France, May 1990.

RCL: a Lisp verification system. In: *Proc. Tenth Intl. Conf. on Automated Deduction* (CADE-10, Kaiserslautern, Germany), ed. Mark E. Stickel, LNCS 449, Springer-Verlag, 1990, pp. 659-660.

An interactive enhancement to the Boyer-Moore Theorem Prover. In: *Proc. 9th Intl. Conf. on Automated Deduction* (CADE-9, Argonne, Illinois, 23-26 May 1988), ed. E. Lusk and R. Overbeek, LNCS 310, Springer-Verlag, Berlin, 1988, pp. 735-736.

Comparing Specification Paradigms for Secure Systems: Gypsy and the Boyer-Moore Logic (with William Young), in: *Proceedings of the 10th National Computer Security Conference (1987).*

Remarks on Weak Notions of Saturation in Models of Peano Arithmetic (with J. Schmerl). *Journal of Symbolic Logic* 52 (1987), pp. 129-148.

The Hanf Number of Stationary Logic (with S. Shelah). *Notre Dame J. Formal Logic* 27 (1986), pp. 111-123.

A Note on the Hanf Number of Second-Order Logic. *Notre Dame J. Formal Logic* 26 (1985), 305-308.

The Quantifier "There Exist Uncountably Many" and Some of Its Relatives, in: *Model-Theoretic Logics* (J. Barwise and S. Feferman, editors), Springer-Verlag, 1985, pp. 123-176.

On Random Models of Finite Power and Monadic Logic (with S. Shelah). *Discrete Mathematics* 54 (1985), pp. 285-293.

A Prototype Theorem-Prover for a Higher-Order Functional Language (with R. Boyer). In *Proceedings of VERkshop III — a formal verification workshop*, Watsonville, CA, February 1985. ACM SIGSOFT Software Engineering Notes, Volume 10, Issue 4, August 1985, ACM, New York, NY, USA.

On Expandability of Models of Arithmetic and Set Theory to Models of Weak Second-Order Theories. *Fund. Math.* CXXII (1984), pp. 57-60.

Some Remarks on Equivalence in Infinitary and Stationary Logic. *Notre Dame J. Formal Logic* 25 (1984), pp. 383-389.

Saturation and Simple Extensions of Models of Peano Arithmetic (with J. Schmerl), *Ann. Pure and Applied Logic* 27 (1984), pp. 109-136.

The Strength of Nonstandard Methods in Arithmetic (with C.W. Henson and H.J. Keisler). *J. Symbolic Logic* 49 (1984), pp. 1039-1058.

A Nonconservativity Result on Global Choice (with S. Shelah). *Ann. Pure and Applied Logic* 27 (1984), pp. 209-214.

Mutually Generic Classes and Incompatible Expansions. *Fund. Math.* CXXI (1984), pp. 213-218.

Definable Ultrapowers and Ultrafilters over Admissible Ordinals (with E. Kranakis). *Zeitschr. f. Math. Logik und Grund. d. Math.* 30 (1984), pp. 97-118.

Filter Logics on ω. *J. Symbolic Logic* 49 (1984), pp. 241-256.

Blunt and Topless End Extensions of Models of Set Theory. *J. Symbolic Logic* 48 (1983), pp. 1053-1071.

Set Theory with a Filter Quantifier. *J. of Symbolic Logic* 48 (1983), pp. 263-287.

On Existence of Σ_n End Extensions, in; *Logic Year 1979-80, The University of Connecticut*, Lec. Notes Math. 859, Springer-Verlag (1981), pp. 92-103.

Σ_1-Well-founded Compactness (with N. Cutland). *Ann. Math. Logic* 18 (1980), pp. 271-296.

Filter logics: Filters on ω_1. *Ann. Math. Logic* 20 (1980), pp. 155-200.

A New Omitting Types Theorem for L(Q). *J. Symbolic Logic* 44 (1979), pp. 217-231.

Stationary Logic (with J. Barwise and M. Makkai). *Ann. Math. Logic* 13 (1978), pp. 171-224.

A Correction to "Stationary Logic" (with J. Barwise and M. Makkai). *Ann. Math. Logic* 20 (1981), pp. 231-232.

A Rather Classless Model. *Proceedings Amer. Math. Soc.* 62 (1977), pp. 330-333.

Internal Notes - EDS, Inc.

NOTE: I have omitted here those technical reports that are listed above among my publications.

A Proposal for COGEN Configuration Management (DRAFT) (with W. Bevier). Note 4. October, 1997.

Generic Rules Document: Temporary expansion, Version 1.3. Note 18. February, 1998.

Generic Rules Document: Temporary expansion, Version 1.8. Note 19. July, 1998.

Procedures for Testing Cogen. Note 23. March, 1998.

A Proposal for Cogen Source and Release Directories. Note 25. Feb., 1998.

Flow Analysis in Cogen. Note 28. March, 1998.

A Proposal for Date Shifting in COGEN 2000 (DRAFT 1.5). Note 30. April, 1998.

Remediation Involving Small Constants. Note 31. April, 1998.

Data Flow in Cogen. Note 35. June, 1998.

Expansion: CC Insertion, Noncompliance, and Datapush. Note 37. August, 1998.

Def-Use in Cogen. Note 39. October, 1998.

Technical Reports - Computational Logic, Inc.

NOTE: I have omitted here those technical reports that are listed above among my publications.

Combining an Interpreter-Based Approach to Software Verification with Verification Condition Generation. Technical Report 97, April, 1994.

(with Bishop Brock and Warren Hunt) The FM9001 Microprocessor Proof. Technical Report 86, December, 1994.

An Assistant for Reading Nqthm Proof Output. Technical Report 85. November, 1992.

Quantification in Nqthm: a Recognizer and Some Constructive Implementations. Technical Report 81. August, 1992.

Introduction to Modeling Computing Systems Using Nqthm (DRAFT). Technical Report 80. July, 1992.

Response to FM91 Survey of Formal Methods: Nqthm and Pc-Nqthm. Technical Report 75. March, 1992.

Addition of Free Variables to an Interactive Enhancement of the Boyer-Moore Theorem Prover. Technical Report 42. May, 1989 (revised March 1990).

(with M. Wilding) A Parallel Version of the Boyer-Moore Prover. Technical Report 39. February, 1989.

A User's Manual for an Interactive Enhancement to the Boyer-Moore Theorem Prover. Technical Report 19. May, 1988.

Technical Reports - Institute for Computing Science and Computer Applications, Univ. of Texas at Austin

Comparing Gypsy and the Boyer-Moore Logic for Specifying Secure Systems (with William D. Young). Technical Report 59. May, 1987.

A User's Manual for an Interactive Enhancement to the Boyer-Moore Theorem prover. Technical Report 60. August, 1987.

Technical Reports - Austin Research Center

ARC 86-08 Summary of Evaluation Policy Studies

ARC 86-01 A Sound Theorem-Prover for a Higher-Order Functional Language

ARC 85-12 A Mechanically-Checked Proof of Correctness of a SASL Unification Program

ARC 85-11 A Mechanically-Checked Proof of Correctness of a SASL Pattern Matching Program

ARC 85-10 A Mechanically-Checked Proof of Correctness of a SASL Quicksort Program

ARC 85-08 A Basic Introduction to the Boyer-Moore System and its Modification for SASL

ARC 85-03 Syntax, Semantics, and a Formal Logic for SASL (with D. Surber)

ARC 84-14 A Direct Construction of a SASL Domain

ARC 84-16 On the Feasibility of Mechanically Verifying SASL Programs (with R. Boyer)

— — —

Last revised: March 3, 2023