References

[1] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development: Coq'Art: The Calculus of Inductive Constructions*. Texts in Theoretical Computer Science. Springer, 2004.

[2] Samuel Boutin. Using reflection to build efficient and certified decision procedures. In *Proceedings of the Third International Symposium on Theoretical Aspects of Computer Software*, pages 515–529, 1997.

[3] Robert S. Boyer, Matt Kaufmann, and J Strother Moore. The Boyer-Moore theorem prover and its interactive enhancement. *Computers and Mathematics with Applications*, 29(2):27–62, 1995.

[4] Venanzio Capretta. General recursion via coinductive types. *Logical Methods in Computer Science*, 1(2):1–18, 2005.

[5] Adam Chlipala. Parametric higher-order abstract syntax for mechanized semantics. In *Proceedings of the 13th ACM SIGPLAN International Conference on Functional Programming*, pages 143–156, 2008.

[6] Adam Chlipala. A verified compiler for an impure functional language. In *Proceedings of the 37th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 93–106, 2010.

[7] Coq Development Team. The Coq proof assistant reference manual, version 8.4. 2012.

[8] Thierry Coquand. An analysis of Girard's paradox. In *Proceedings of the Symposium on Logic in Computer Science*, pages 227–236, 1986.

[9] Thierry Coquand and Gérard Huet. The Calculus of Constructions. *Information and Computation*, 76(2-3), 1988.

[10] H. B. Curry. Functionality in combinatory logic. *Proceedings of the National Academy of Sciences of the United States of America*, 20(11):584–590, 1934.

[11] Nicolas G. de Bruijn. Lambda-calculus notation with nameless dummies: A tool for automatic formal manipulation with application to the Church-Rosser theorem. *Indagationes Mathematicae*, 34(5):381–392, 1972.

[12] Eduardo Giménez. A tutorial on recursive types in Coq. Technical Report 0221, INRIA, May 1998.

[13] Georges Gonthier. Formal proof—the four-color theorem. *Notices of the American Mathematical Society*, 55(11):1382–1393, 2008.

[14] William A. Howard. The formulae-as-types notion of construction. In Jonathan P. Seldin and J. Roger Hindley, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 479–490. Academic Press, 1980. Original paper manuscript from 1969.

[15] Gérard Huet. The undecidability of unification in third order logic. *Information and Control*, pages 257–267, 1973.

[16] John Hughes. Why functional programming matters. *Computer Journal*, 32:98–107, 1984.

[17] Matt Kaufmann, Panagiotis Manolios, and J Strother Moore. *Computer-Aided Reasoning: An Approach*. Kluwer Academic Publishers, 2000.

[18] Gerwin Klein, Kevin Elphinstone, Gernot Heiser, June Andronick, David Cock, Philip Derrin, Dhammika Elkaduwe, Kai Engelhardt, Rafal Kolanski, Michael Norrish, Thomas Sewell, Harvey Tuch, and Simon Winwood. seL4: Formal verification of an OS kernel. In *Proceedings of the 22nd ACM Symposium on Operating Systems Principles*, pages 207–220, 2009.

[19] Xavier Leroy. A formally verified compiler back-end. *Journal of Automated Reasoning*, 43(4):363–446, 2009.

[20] Xavier Leroy and Hervé Grall. Coinductive big-step operational semantics. In *Proceedings of the 15th European Symposium on Programming*, pages 54–68, 2006.

[21] John W. Lloyd. *Foundations of Logic Programming*. Springer, 2nd edition, 1987.

[22] David MacQueen. Modules for Standard ML. In *Proceedings of the 1984 ACM Symposium on LISP and Functional Programming*, pages 198–207, 1984.

[23] Conor McBride. Elimination with a motive. In *Proceedings of the International Workshop on Types for Proofs and Programs*, pages 197–216, 2000.

[24] Adam Megacz. A coinductive monad for Prop-bounded recursion. In *Proceedings of the ACM Workshop Programming Languages Meets Program Verification*, pages 11–20, 2007.

[25] J Strother Moore. *Piton: A Mechanically Verified Assembly-Level Language*. Automated Reasoning Series. Kluwer Academic Publishers, 1996.

[26] J Strother Moore, Tom Lynch, and Matt Kaufmann. A mechanically checked proof of the correctness of the kernel of the AMD5k86 floating-point division algorithm. *IEEE Transactions on Computers*, 47(9):913–926, 1998.

[27] George C. Necula. Proof-carrying code. In *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 106–119, 1997.

[28] Zhaozhong Ni and Zhong Shao. Certified assembly programming with embedded code pointers. In *Proceedings of the 33rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 320–333, 2006.

[29] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL—A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lecture Notes in Computer Science*. Springer, 2002.

[30] Chris Okasaki. Red-black trees in a functional setting. *Journal of Functional Programming*, 9:471–477, 1999.

[31] Christine Paulin-Mohring. Inductive definitions in the system Coq—rules and properties. In *Proceedings of the International Conference on Typed Lambda Calculi and Applications*, pages 328–345, 1993.

[32] Lawrence C. Paulson. *Isabelle: A Generic Theorem Prover*, volume 828 of *Lecture Notes in Computer Science*. Springer, 1994.

[33] Simon Peyton Jones, Lennart Augustsson, Dave Barton, Brian Boutel, Warren Burton, Joseph Fasel, Kevin Hammond, Ralf Hinze, Paul Hudak, John Hughes, Thomas Johnsson, Mark Jones, John Launchbury, Erik Meijer, John Peterson, Alastair Reid, Colin Runciman, and Philip Wadler. *Haskell 98 Language and Libraries: The Revised Report*. 1998. Section 4.3.3.

[34] Simon L. Peyton Jones and Philip Wadler. Imperative functional programming. In *Proceedings of the 20th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 71–84, 1993.

[35] Frank Pfenning and Conal Elliott. Higher-order abstract syntax. In *Proceedings of the ACM SIGPLAN 1988 Conference on Programming Language Design and Implementation*, pages 199–208, 1988.

[36] Benjamin C. Pierce. *Types and Programming Languages*. MIT Press, 2002.

[37] Benjamin C. Pierce. *Types and Programming Languages*. MIT Press, 2002. Section 9.4.

[38] John C. Reynolds. Types, abstraction, and parametric polymorphism. *Information Processing*, pages 513–523, 1983.

[39] John C. Reynolds. The discoveries of continuations. *Lisp and Symbolic Computation*, 6(3-4):233–248, November 1993.

[40] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proceedings of the IEEE Symposium on Logic in Computer Science*, pages 55–74, 2002.

[41] John Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, January 1965.

[42] Leon Sterling and Ehud Shapiro. *The Art of Prolog.* MIT Press, 2nd edition, 1994.

[43] Thomas Streicher. *Semantical Investigations into Intensional Type Theory.* Habilitationsschrift, LMU München, 1993.

[44] Philip Wadler. The essence of functional programming. In *Proceedings of the 19th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 1–14, 1992.

[45] Philip Wadler and Stephen Blott. How to make ad-hoc polymorphism less ad hoc. In *Proceedings of the 16th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 60–76, 1989.

[46] Geoffrey Washburn and Stephanie Weirich. Boxes go bananas: Encoding higher-order abstract syntax with parametric polymorphism. *Journal of Functional Programming*, 18(1):87–140, 2008.

[47] Markus Wenzel. Isar—A generic interpretative approach to readable formal proof documents. In *Proceedings of the 12th International Conference on Theorem Proving in Higher Order Logics*, pages 167–184, 1999.

[48] Benjamin Werner. Sets in types, types in sets. In *Proceedings of the Third International Symposium on Theoretical Aspects of Computer Software*, pages 530–546, 1997.

[49] Glynn Winskel. *The Formal Semantics of Programming Languages.* MIT Press, 1993. Chapter 8.

[50] Hongwei Xi, Chiyan Chen, and Gang Chen. Guarded recursive datatype constructors. In *Proceedings of the 30th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 224–235, 2003.