Theorem Proving and the Real Numbers: Overview and Challenges

Lawrence C. Paulson

Computer Laboratory, University of Cambridge, England
lp15@cl.cam.ac.uk

One of the first achievements in automated theorem proving was Jutting's construction of the real numbers using AUTOMATH [14]. But for years afterwards, formal proofs focused on problems from functional programming and elementary number theory. In the early 90s, John Harrison revived work on the reals by formalising their construction using HOL [8] and by undertaking an extensive programme of research into verifying floating point arithmetic, including the exponential and trigonometric functions [9–11].

MetiTarski represents a different approach to theorem proving about the reals. Reducing everything to first principles is rigorous, but makes proofs of the simplest statements extremely time-consuming. Many other automatic theorem provers are confined to linear arithmetic, or at best, polynomial comparisons. MetiTarski can prove complicated assertions involving transcendental functions. It takes many of their properties as axioms, and reasons from these properties using sophisticated decision procedures. MetiTarski has recently been integrated with other powerful reasoning tools, including KeYmaera [19] and PVS [17]. With this power, proofs involving such things as aircraft manoeuvres and the stability of hybrid systems can be undertaken, even when the dynamics are described by complicated formulas involving many special functions. Examples of this research can be found in these proceedings, for example, Denman's work on qualitative abstraction of hybrid systems [6].

This very success raises the question of how to recover the rigour of LCF-style theorem proving without losing the power of MetiTarski. The standard answer to this question (used by Isabelle's Sledgehammer for example [18]) is for the external prover to generate some sort of certificate that can be checked rigorously. The point is that the expensive proof search does not need to be checked, but only the proof that was actually found.

Checking a certificate using a separate theorem prover, such as Isabelle, requires machine formalisations of all the underlying mathematics. Since Harrison's work mentioned above, researchers worldwide have formalised substantial chunks of real analysis, including measure theory and probability theory [12, 16]. Independently, from the 1960s onwards, computer algebra systems enjoyed rapid development, as did decision procedures for real arithmetic. Much recent work has focused on formalising computer algebra algorithms within theorem provers, especially Coq [2, 15]. Investigations into special function inequalities have been conducted using PVS [5].

Nevertheless, the mathematics needed to certify the sort of proofs found by MetiTarski does not appear to have been formalised as yet. MetiTarski relies on an external decision procedure for *real-closed fields* (RCF) [7] to test the satisfiability of first-order formulas involving polynomials. The underlying algorithm is called CAD (Cylindrical Algebraic Decomposition) and QEPCAD [3] is a well-known implementation, although it has also been implemented in Mathematica and Z3 [13]. Each of these implementations is very complicated, and there is no obvious way to verify their results.

The underlying mathematics is real algebraic geometry [1]. MetiTarski also relies upon upper and lower bounds for the fractions it reasons about, given in the form of truncated power series or rational functions derived from continued fractions [4]. The necessary mathematics here belongs to approximation theory, and unusually, we are not concerned with the closeness of the approximations; the soundness of MetiTarski relies only upon the property that they are indeed upper or lower bounds. Proving these properties formally appears to require a substantial effort. And although we are only concerned with the real numbers, the necessary theory is most easily reached via complex analysis. That branch of mathematics remains largely unformalised at the moment, so we have much to do.

Acknowledgements. The Edinburgh members of the project team are Paul Jackson, Grant Passmore and Andrew Sogokon. The Cambridge team includes James Bridge, William Denman and Zongyan Huang. We are grateful to our outside collaborators such as César Muñoz, Eva Navarro-López, André Platzer, and others not listed here.

The research was supported by the Engineering and Physical Sciences Research Council [grant numbers EP/I011005/1, EP/I010335/1].

References

1. Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry.* Springer, 2nd edition, 2006.
2. Yves Bertot, Frédérique Guilhot, and Assia Mahboubi. A formal study of Bernstein coefficients and polynomials. *Mathematical Structures in Computer Science*, 21(04):731–761, 2011.
3. Christopher W. Brown. QEPCAD B: a program for computing with semi-algebraic sets using CADs. *SIGSAM Bulletin*, 37(4):97–108, 2003.
4. A. Cuyt, V. Petersen, B. Verdonk, H. Waadeland, and W.B. Jones. *Handbook of Continued Fractions for Special Functions.* Springer, 2008.
5. Marc Daumas, César Muñoz, and David Lester. Verified real number calculations: A library for integer arithmetic. *IEEE Trans. Computers*, 58(2):226–237, 2009.
6. William Denman. Verifying nonpolynomial hybrid systems by qualitative abstraction and automated theorem proving. In *NASA Formal Methods, 6th International Symposium, NFM 2014*, 2014. These proceedings.
7. Andreas Dolzmann, Thomas Sturm, and Volker Weispfenning. Real quantifier elimination in practice. In B.Heinrich Matzat, Gert-Martin Greuel, and Gerhard Hiss, editors, *Algorithmic Algebra and Number Theory*, pages 221–247. Springer, 1999.

8. John Harrison. Constructing the real numbers in HOL. *Formal Methods in System Design*, 5:35–59, 1994.
9. John Harrison. Floating point verification in HOL Light: the exponential function. *Formal Methods in System Design*, 16:271–305, 2000.
10. John Harrison. Formal verification of floating point trigonometric functions. In Jr. Hunt, Warren A. and Steven D. Johnson, editors, *Formal Methods in Computer-Aided Design*, LNCS 1954, pages 254–270. Springer, 2000.
11. John Harrison. Formal verification of IA-64 division algorithms. In Mark Aagaard and John Harrison, editors, *Theorem Proving in Higher Order Logics: TPHOLs 2000*, LNCS 1869, pages 233–251. Springer, 2000.
12. Joe Hurd. Verification of the Miller-Rabin probabilistic primality test. *Journal of Logic and Algebraic Programming*, 56:3–21, 2002.
13. Dejan Jovanovic and Leonardo Mendonça de Moura. Solving non-linear arithmetic. In Bernhard Gramlich, Dale Miller, and Uli Sattler, editors, *IJCAR 2012*, LNCS 7364, pages 339–354. Springer, 2012.
14. L.S. van Benthem Jutting. *Checking Landau's "Grundlagen" in the AUTOMATH System*. PhD thesis, Eindhoven University of Technology, 1977.
15. Assia Mahboubi. Implementing the CAD algorithm within the Coq system. *Mathematical Structure in Computer Sciences*, 17, 2007.
16. Tarek Mhamdi, Osman Hasan, and Sofiène Tahar. Formalization of measure theory and Lebesgue integration for probabilistic analysis in HOL. *ACM Trans. Embedded Comput. Syst.*, 12(1):13, 2013.
17. S. Owre, S. Rajan, J.M. Rushby, N. Shankar, and M.K. Srivas. PVS: Combining specification, proof checking, and model checking. In Rajeev Alur and Thomas A. Henzinger, editors, *Computer Aided Verification: 8th International Conference, CAV '96*, LNCS 1102, pages 411–414. Springer, 1996.
18. Lawrence 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: TPHOLs 2007*, LNCS 4732, pages 232–245. Springer, 2007.
19. André Platzer and Jan-David Quesel. KeYmaera: A hybrid theorem prover for hybrid systems. In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Automated Reasoning — 4th International Joint Conference, IJCAR 2008*, LNCS 5195, pages 171–178. Springer, 2008.