Automated Formal Proofs for Polynomial and Transcendental Problems

Prof. Lawrence C. Paulson
Computer Laboratory, University of Cambridge

1 Previous Research and Track Record

Lawrence C. Paulson is Professor of Computational Logic at the University of Cambridge, where he has held established posts since 1983. One of his main activities is developing proof tools. His early work made fundamental contributions to Prof. M. J. C. Gordon's proof assistant, HOL. In 1986, Paulson introduced Isabelle, a generic proof assistant. Isabelle supports higher-order logic (HOL), Zermelo-Fraenkel set theory (ZF) and other formalisms. Many developments are due to Prof. Tobias Nipkow's group at the Technical University of Munich. Automatic proof search, one of Isabelle's particular strengths, is however due to Paulson [22].

The work will be done within the Cambridge Automated Reasoning Group. Hardware verification was pioneered here by Prof. Gordon and his students. They introduced what have become standard techniques, such as the use of higher-order logic to model hardware and software systems. The group's work continues to attract worldwide attention. Former members such as Dr. John Harrison have taken formal verification to Intel and other companies. The group has built two of the world's leading proof environments, namely HOL and Isabelle. Institutes using Isabelle as a basis for their research include the University of Edinburgh, Carnegie-Mellon University and Australia's Defence Science and Technology Organisation (DSTO). The Verisoft project, which uses Isabelle extensively, comprises 11 partners, including Infineon Technologies and BMW.[1]

The EPSRC has funded several projects at Cambridge involving Isabelle. They include the following:

- *Verifying Electronic Commerce Protocols* (EPSRC ref. GR/R 01156/01), 2000–03. This project had the objective of verifying security protocols of industrial complexity. The huge SET protocol suite was analysed and some vulnerabilities found. The research assistant, Giampaolo Bella, investigated the Zhou-Gollmann non-repudiation protocol and a certified electronic mail protocol designed by Abadi et al. This project produced numerous publications.

- *Automation for Interactive Proof* (EPSRC ref. GR/S57198/01), 2004–07. This project gave interactive proof tools improved automation through an

[1] `http://www.verisoft.de/ProjectConsortium.html`

effective combination of interactive and automatic tools. It developed techniques for transferring subgoals from an interactive prover to an automatic one and for transferring proofs in the opposite direction. A complete working system was built, linking Isabelle to the resolution provers E, SPASS and Vampire. The hardest phase concerned reconstruction of the proofs within Isabelle.

- *Beyond Linear Arithmetic: Automatic Proof Procedures for the Reals* (EPSRC ref. EP/C013409/1), 2005–08. This project is investigating advanced methods of proving theorems about the transcendental functions: log, exp, sin, cos, etc. [2]. A prototype prover has been constructed [1], combining a resolution theorem prover (Metis) with a decision procedure for the theory of Real Closed Fields (QEPCAD) and axioms concerning the elementary functions. This prover can already prove numerous problems involving log and exp, with runtimes typically below one second.

This last project is the main inspiration for the new proposal. We now have automatic techniques for proving complicated assertions involving the reals. The use of a powerful decision procedure (QEPCAD) is crucial to this work, but unfortunately its output is not a proof in the traditional sense. How do we prove statements about the reals within an interactive verification environment such as Isabelle, achieving both automation and soundness? The key point is to identify practicable and verifiable techniques that cover a variety of common special cases. The two projects are complementary, one using general methods (resolution and QEPCAD), the other using a selection of highly focused methods.

The designated Research Assistant, Amine Chaieb, is currently completing his PhD at the Technical University of Munich. He is an experienced Isabelle developer, with the detailed knowledge necessary to implement reflected proof procedures within Isabelle. He also has a through knowledge of the mathematical theories relevant to the project. He has already implemented proof procedures that are similar in spirit to those proposed below, such as a decision procedure for ordered fields. He has experience of using computer algebra systems to increase proof automation, for example applying Gröbner bases to universal problems over rings.

2 Description of Proposed Research

The proposal is to investigate methods for automating formal proofs about polynomials and transcendental functions, and to implement them within an interactive theorem prover, Isabelle. The problems are formulated as universal polynomial or transcendental properties or even as general first order arithmetic formulae. We expect a tremendous increase in automation for the universal case and for several cases of the general first order case as well. We also expect improved automation for proofs about power series and transcendental functions.

2.1 Background

Interactive proof tools such as Isabelle and HOL are widely used for verification, even in industry. Their reliance on the LCF proof architecture ensures soundness, and they have many facilities for managing complex specifications and proofs. However, they can be difficult to use; for example, proving even obvious facts involving the real numbers can take days.

The problems we want to solve are inherently non-linear and arise in many applications and formalizations of mathematics [4], algorithm and program verification [13], hardware implementations of transcendental functions, electrical engineering (circuit networks [27]), control theory, scheduling, robotics [18] and many other fields. Although we know since Tarski [23] that the first order theory of real closed fields admits quantifier elimination and is decidable, the intractable complexity of the problem and all known methods make it almost useless. The only procedures seriously integrated (by yielding a proof of their result) in an LCF theorem prover are very simple [12, 15, 17, 20]; they have non-elementary complexity bounds, making them useless even for simple examples. Sophisticated algorithms like cylindrical algebraic decomposition (CAD) seem to be a serious challenge for theorem provers and might require several years to verify [19].

Our approach is to design and implement efficient methods for special and practical cases, and to have one simple complete algorithm as a fall-back solution to eliminate one quantifier. Chaieb [9] has almost finished the implementation of Cohen's procedure [12]. The techniques we want to employ can be classified in two categories: theorem proving and mathematical techniques.

Theorem proving

We shall use *computational reflection* for several subproblems, along with a certificate-based integration of external tools. Reflection is a major strength of higher order logic: we formalize proof procedures inside an *executable* fragment of the logic and prove them correct. For subsequent applications, the prover simply executes ML code generated from the formalizations. This approach has been used with great success in the Coq system [24]. Reflection is also available in Isabelle, and we have already used this mechanism for serious examples: quantifier elimination for the first-order theory of linear arithmetic over the reals, the integers and their combination [6, 8, 10]. Another application is to check ideal membership certificates generated by computer algebra systems [9]. Using

this technique, we achieved a speed-up by a factor of 200, compared with a traditional proof-producing procedure.

For the certificate-based approach, we identify subproblems that have efficiently checkable certificates and tools that can generate such certificates. For instance, proving that a polynomial has a sum of squares representation yields a simple proof of its non-negativity. The integration of such procedures then reduces to implementing a checker for the certificates. We successfully employed this technique to implement proof procedures for universal statements over rings (no ordering) using Gröbner Bases and Hilbert's Nullstellensatz together with computer algebra systems generating ideal membership certificates and to prove large numbers prime [9, 11]. We also intend to combine reflection and the certificates approach: find a certificate by external tools and check it by a verified, reflected checker. We expect our approach to scale up to serious applications.

Mathematical techniques

Since the general methods tend to be inefficient, we intend to investigate some practically (and also mathematically) interesting special cases.

For universal equations and inequations, we propose to use Gröbner bases. This is not complete over the reals, but most problems in practice hold in all rings or fields. This technique is already available in Isabelle.

For universal inequalities, we can use the efficient semi-definite programming (SDP) tools to obtain a Stengle's Positivstellensatz refutation [16, 21]. Given a formula

$$F = \bigwedge_{i=1}^{n} p_i(\vec{x}) = 0 \wedge \bigwedge_{i=1}^{m} q_i(\vec{x}) \geq 0 \wedge \bigwedge_{i=1}^{t} r_i(\vec{x}) \neq 0,$$

Stengle's Positivstellensatz guarantees that this statement is false over the reals if and only if there exist $p = \sum_{i=1}^{n} a_i p_i$ and q in the cone of $q_1, \ldots, q_m$ and $r = \prod_{i=1}^{t} r_i^{k_i}$, for some $a_1, \ldots, a_n$ and $k_1, \ldots, k_t$, such that $p + q + r^2 = 0$, which obviously contradicts F. Finding such a refutation can be formulated as a semidefinite program [21].

Problems with quantifiers, where the quantified variables only occur linearly, fall into the case of parametric linear arithmetic. It is efficiently solvable using established methods [14, 18]. This already covers many interesting applications in electrical engineering, transportation problems and expert systems [18, 27]. More generally, if the quantified variable occurs only in polynomials with degree $n \leq 4$, then we can use the standard techniques for solving an algebraic equation of degree n to eliminate the quantifier [25, 26]. This approach is much more efficient than generic quantifier elimination methods, but it is not applicable if $n > 4$ due the Abel-Ruffini theorem on the unsolvability of the general polynomial equation.

For *univariate* universal problems, a naive use of SDP can be unsatisfactory, especially if the coefficients are ratios of large integers. This is the case for power series and their approximations. For this case, we know that any positive semidefinite univariate polynomial is the sum of two squares. Here we intend to

use computer algebra systems to approximate complex roots and compute square-free decompositions of (possibly perturbated) polynomials, which already gives a squares part. For the irreducible parts, we use SDP.

The first order theories of mixed linear arithmetic with a transcendental function (such as the exponential, hyperbolic functions or arctan) and simple trigonometric functions (sin applied to a polynomial) are decidable. In contrast to previous developments, Weispfenning [3, 28] does not rely on Shnuel's conjecture but uses only Lindemann's theorem. The procedures and Lindemann's theorem itself are challenging in the context of a theorem prover. We consider this last part as follow-on research, but should tackle it if we have time. Problems of this kind occur frequently in formalizations [4].

If none of these cases are applicable, then we fall back to a complete procedure. We have almost finished an implementation of Cohen's quantifier elimination procedure [12] in Isabelle [9]. We aim to eliminate only as many quantifiers as are needed to reduce the problem to a tractable case.

2.2 Programme and Methodology

We envisage the following workplan for the 12 months. We have already done preliminary investigations for this work. Note that some of the tasks will take place concurrently.

Task 1: Theoretical issues and investigation of external tools.

To begin with, we shall develop the proof algorithms and investigate their soundness and completeness for certain relevant subclasses. The problems we address here are the universal univariate case for polynomials and methods to automate reasoning about power series and transcendental functions. At the same time, we shall investigate computer algebra systems and other software to solve the resulting subproblem of the algorithms and their way of interaction. It is important to us that the theoretical study yields an efficient implementation. We already have a modest experience with Singular, PARI/GP, and M2. *(months 1–4)*

Task 2: Proof infrastructure in Isabelle.

To build proof procedures, we shall need to prove a body of theorems (in Isabelle) in order to argue that the transformations are correct in the underlying logical calculus. This part is the formalization of the previous theoretical investigations and constitutes the logical core of the work. Note also that for reflection-based methods there is no difference between proof and implementation. *(months 4–10)*

Task 3: Implementation and integration.

Finally, we shall implement all our results and integrate them into Isabelle. This is a rather technical part, but we already have experience with the Isabelle internals as system developers. The new abstract specification mechanisms in

Isabelle [5] are highly flexible, but require care during implementation. We shall integrate most methods as abstractly as possible, making them available for a class of structures including the reals as a special case. For example, a sum of squares yields non-negativity in every ordered semiring, and not only over the reals. We successfully used this technique [9, 11] for universal and interesting existential problems over commutative rings. This kind of integration meets the expectations of mathematicians and yields maximum generality. For example, our methods will apply to the nonstandard real numbers with no additional effort. *(months 8–12)*

2.3 Relevance to Beneficiaries

The beneficiaries are chiefly in the research world, but the types of problems we intend to solve have many industrial applications.

- The formal methods and theorem proving community will benefit from the techniques we develop. Our formalized theories and procedures will be valuable in their own right. Developers of other proof tools, such as HOL and PVS, will be able to adapt many of them for their purposes.
- Practitioners of engineering, control theory and scheduling (among other fields) will be able to apply our methods for verifying mathematical assertions [18, 27].
- Practitioners of program analysis and verification will benefit directly from our Isabelle implementation, in order to prove verification conditions, properties of control flow or termination [13]. The results should easily surpass our previous work [7, 29].

2.4 Dissemination and Exploitation

The new proof tools will be included in Isabelle and thereby distributed via the Internet under an open source license. Technical and theoretical findings will be presented at conferences and published in academic journals.

References

[1] B. Akbarpour and L. Paulson. Extending a resolution prover for inequalities on elementary functions. In *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 47–61, 2007.

[2] B. Akbarpour and L. C. Paulson. Towards automatic proofs of inequalities involving elementary functions. In B. Cook and R. Sebastiani, editors, *PDPAR: Pragmatics of Decision Procedures in Automated Reasoning*, pages 27–37, 2006.

[3] H. Anai and V. Weispfenning. Deciding linear-trigonometric problems. In *ISSAC '00: Proceedings of the 2000 international symposium on Symbolic and algebraic computation*, pages 14–22. ACM, 2000.

[4] J. Avigad, K. Donnelly, D. Gray, and P. Raff. A formally verified proof of the prime number theorem. *ACM Trans. Comput. Log.*, 9(1), 2007.

[5] C. Ballarin. Interpretation of locales in Isabelle: Theories and proof contexts. In J. M. Borwein and W. M. Farmer, editors, *Mathematical Knowledge Management (MKM 2006)*, LNAI 4108, 2006.

[6] A. Chaieb. Mechanized quantifier elimination for linear real-arithmetic in isabelle/hol. Technical report, Technische Universität München, 2006.

[7] A. Chaieb. Proof-producing program analysis. In K. Barkaoui, A. Cavalcanti, and A. Cerone, editors, *Theoretical Aspects of Computing - ICTAC 2006, Third International Colloquium, Tunis, Tunisia, November 20-24, 2006, Proceedings*, volume 4281 of *Lect. Notes in Comp. Sci.*, pages 287–301. Springer-Verlag, 2006.

[8] A. Chaieb. Verifying mixed real-integer quantifier elimination. In U. Furbach and N. Shankar, editors, *IJCAR*, volume 4130 of *Lect. Notes in Comp. Sci.*, pages 528–540. Springer-Verlag, 2006.

[9] A. Chaieb. *Automated methods for formal proofs in simple arithmetics and algebra*. PhD thesis, Technische Universität München, Germany, 2008. Forthcoming.

[10] A. Chaieb and T. Nipkow. Verifying and reflecting quantifier elimination for Presburger arithmetic. In G. Stutcliffe and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*, volume 3835 of *Lect. Notes in Artif. Int.* Springer-Verlag, 2005.

[11] A. Chaieb and M. Wenzel. Context aware calculation and deduction — ring equalities via Gröbner Bases in Isabelle. In M. Kauers, M. Kerber, R. Miner, and W. Windsteiger, editors, *Towards Mechanized Mathematical Assistants (CALCULEMUS 2007 and MKM 2007)*, LNAI 4573. Springer-Verlag, 2007.

[12] P. Cohen. Decision procedures for real and p-adic fields. *Communications in Pure and Applied Mathematics*, 22:131–151, 1969.

[13] P. Cousot. Proving program invariance and termination by parametric abstraction, lagrangian relaxation and semidefinite programming. In *Sixth International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'05)*, pages 1–24, Paris, France, LNCS 3385, Jan. 2005. Springer-Verlag.

[14] J. Ferrante and C. Rackoff. A decision procedure for the first order theory of real addition with order. *SIAM J. Comput.*, 4(1):69–76, 1975.

[15] J. Harrison. *Theorem Proving with the Real Numbers*. PhD thesis, University of Cambridge, 1996.

[16] J. Harrison. Verifying nonlinear real formulas via sums of squares. In K. Schneider and J. Brandt, editors, *Proceedings of the 20th International Conference on Theorem Proving in Higher Order Logics TPHOLs 2007*,

volume 4732 of *Lect. Notes in Comp. Sci.*, pages 102–118, Kaiserslautern, Germany, 2007. Springer-Verlag.

[17] G. Kreisel and J. Krivine. *Elements of Mathematical Logic (Model theory)*. Studies in Logic and the foundations of mathematics. North-Holland, Amsterdam, 1967.

[18] R. Loos and V. Weispfenning. Applying linear quantifier elimination. *Comput. J.*, 36(5):450–462, 1993.

[19] A. Mahboubi. *Contributions à la certification des calculs dans* **R***: théorie, preuves, programmation*. PhD thesis, Université de Nice Sophia Antipolis, France, 2006.

[20] S. McLaughlin and J. Harrison. A proof-producing decision procedure for real arithmetic. In R. Nieuwenhuis, editor, *CADE-20: 20th International Conference on Automated Deduction, proceedings*, volume 3632 of *Lect. Notes in Comp. Sci.*, pages 295–314, Tallinn, Estonia, 2005. Springer-Verlag.

[21] P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Math. Program*, 96(2):293–320, 2003.

[22] L. C. Paulson. A generic tableau prover and its integration with Isabelle. *Journal of Universal Computer Science*, 5(3), 1999.

[23] A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, 1951.

[24] K. N. Verma, J. Goubault-Larrecq, S. Prasad, and S. Arun-Kumar. Reflecting BDDs in Coq. In J. He and M. Sato, editors, *Advances in Computing Science—ASIAN 2000: 6th Asian Computing Science Conference*, LNCS 1961, pages 162–181. Springer-Verlag, 2000.

[25] V. Weispfenning. Quantifier elimination for real algebra — the cubic case. In *International Symposium on Symbolic and Algebraic Computation*, pages 258–263, 1994.

[26] V. Weispfenning. Quantifier elimination for real algebra: the quadratic case and beyond. *Applicable Algebra in Engineering, Communication and Computing*, 8(2):85–101, 1997.

[27] V. Weispfenning. Simulation and optimization by quantifier elimination. *Journal of Symbolic Computation*, 24(2):189–208, 1997.

[28] V. Weispfenning. Deciding linear-exponential problems. *SIGSAM Bulletin (ACM Special Interest Group on Symbolic and Algebraic Manipulation)*, 34(1):30–31, 2000.

[29] M. Wildmoser, A. Chaieb, and T. Nipkow. Bytecode analysis for proof carrying code. *Proceedings of the 1st Workshop on Bytecode Semantics, Verification and Transformation, Electronic Notes in Computer Science*, 2005.