Mechanising Temporal Reasoning: Summary

Lawrence C. Paulson and Michael J. C. Gordon

The project, funded by the Engineering and Physical Sciences Research Council (EPSRC), was undertaken to continue research and development involving the proof tool Isabelle. The emphasis was on temporal logics. Its results include a mechanisation of the UNITY formalism [6] and an automatic tableaux-based proof tactic, `Blast_tac`. Both of these are distributed with Isabelle. The research assistants were C. Owens and G. Bella, who published work on the temporal properties of security protocols [2].

Our UNITY mechanisation uses a relational semantics. The full theory of UNITY's safety and progress properties has been developed from first principles, including difficult theorems such as PSP (Progress-Safety-Progress) and Completion. Many proofs from the UNITY literature were mechanised, including two-process mutual exclusion and Andersen's lift example. We have also mechanised elements of recent theories on reasoning about program composition, such as the *guarantees* relation.

`Blast_tac` [5] consists of a tableau theorem prover coded directly in ML; for greater speed, it bypasses Isabelle's proof engine. If it finds a proof then it issues a string of tactics that Isabelle applies to prove the goal; `Blast_tac` therefore cannot cause unsoundness. Like Isabelle's other tools, `Blast_tac` is *generic*: it works with any suitable rules supplied by the user rather than with the fixed rules of predicate logic.

PhD students in the Isabelle group did outstanding work. C. Ballarin's thesis concerns integrating computer algebra with theorem proving; he integrates Isabelle with a library of computer algebra algorithms [1]. J. Fleuriot's has formalized non-standard analysis using an ultrafilter construction. Combining this with an axiomatic framkework for geometry, he has mechanised proofs from Newton's *Principia* and shown them to be rigorous [3] despite their reliance on notions such as "infinitely close." F. Kammüller has investigated modularity in proof tools; he has demonstrated his work by proving substantial results of algebra, such as Sylow's theorem [4].

References

[1] C. Ballarin and L. C. Paulson. A pragmatic approach to extending provers by computer algebra — with applications to coding theory. *Fundamenta Informaticae*, 39:1–20, 1999.

[2] G. Bella and L. C. Paulson. Kerberos version IV: Inductive analysis of the secrecy goals. In J.-J. Quisquater, Y. Deswarte, C. Meadows, and D. Gollmann, editors, *Computer Security — ESORICS 98*, LNCS 1485, pages 361–375. Springer, 1998.

[3] J. D. Fleuriot and L. C. Paulson. A combination of nonstandard analysis and geometry theorem proving, with application to Newton's Principia. In C. Kirchner and H. Kirchner, editors, *Automated Deduction — CADE-15 International Conference*, LNAI 1421, pages 3–16. Springer, 1998.

[4] F. Kammüller and L. C. Paulson. A formal proof of Sylow's theorem: An experiment in abstract algebra with Isabelle HOL. *Journal of Automated Reasoning*, 23:235–264, 1999.

[5] L. C. Paulson. A generic tableau prover and its integration with Isabelle. *Journal of Universal Computer Science*, 5(3):73–87, 1999.

[6] L. C. Paulson. Mechanizing UNITY in Isabelle. *ACM Transactions on Computational Logic*, 1(1):3–32, 2000.