Little (but Hard) Theorems about Big Systems: Some Case Studies

J Strother Moore

Department of Computer Sciences, University of Texas at Austin

1 abstract

Today it is impractical to prove – formally and mechanically – the correctness of entire computing systems of commercial interest. There are a variety of reasons for this, both technical and economic. However, mechanized theorem proving is nevertheless relevant in commercial hardware and software production. But practical considerations require that we focus our attention on problems that are both technically feasible within the time limits available and of interest to system designers.

In this series of lectures I will describe several such "little theorems," that is, theorems that address issues of concern to designers without trying to address the complete correctness of the system. The theorems have all been formalized and proved with the ACL2 theorem prover. "ACL2" stands for A Computational Logic for Applicative Common Lisp. It is a theorem prover in the Boyer-Moore tradition that uses rewriting, decision procedures, mathematical induction and a variety of other proof techniques to prove theorems in a first-order mathematical theory of recursive functions and inductively constructed objects.

After a sketch of ACL2 [2], I will present proofs about a comparator sort algorithm underlying a microcode program in a Motorla digital signal processor [1], the correctness of various floating-point operations on AMD microprocessors [4, 5], and the properties of an executable formal model of the Java Virtual Machine [3].

Keywords: ACL2, hardware verification, software verification, theorem proving.

References

1. B. Brock and J S. Moore. A mechanically checked proof of a comparator sort algorithm, 1999. `http://-www.cs.utexas.edu/users/moore/publications/csort/main.ps.Z`.
2. M. Kaufmann, P. Manolios, and J S. Moore. *Computer-Aided Reasoning: An Approach.* Kluwer Academic Press, Boston, MA., 2000. http://www.cs.utexas.edu/users/moore/publications/acl2-papers.-html#Books.
3. H. Liu and J S. Moore. Executable JVM model for analytical reasoning: A study. In *Workshop on Interpreters, Virtual Machines and Emulators 2003 (IVME '03)*, San Diego, CA, June 2003. ACM SIGPLAN. http://www.cs.utexas.edu/users/moore/publications/m6.ps.
4. J S. Moore, T. Lynch, and M. 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, September 1998. http://www.cs.utexas.edu/users/moore/publications/div-march-96.ps.
5. D. Russinoff. A mechanically checked proof of IEEE compliance of a register-transfer-level specification of the AMD-K7 floating-point multiplication, division, and square root instructions. *London Mathematical Society Journal of Computation and Mathematics*, 1:148–200, December 1998. http://www.onr.com/-user/russ/david/k7-div-sqrt.html.