Industrial Use of ACL2: Present and Future

David S. Hardin, Ph.D.
Advanced Technology Center
david.hardin@rockwellcollins.com

Rockwell Collins

Our Uses of ACL2 to Date

- Microcode Modeling and Proofs
- AAMP7 Information Flow Proofs (GWV Theorem)
    - NSA MILS Accreditation
- Green Hills Information Flow Proofs (GWVr2 Theorem)
    - EAL6+ Accreditation
- AAMP7 Instruction Set Modeling and Proofs
    - Interface to Eclipse-based Debugger
- MicroCryptol Runtime
- Proofs for Guard Prototype (AAMP7 code, vFAAT)
- Data Flow Logic (DFL) for C code
- LLVM Modeling and Proofs
- Other things we can’t talk about…

Themes:

- ***Automated High-Level Property Verification for Low-Level Artifacts***
- ***Validation Enabled by Executable Formal Models***

My ACL2 “Wish List”

- **Detailed, Executable Formal Models for Common Microprocessors**
  - x86-64, ARM, maybe PowerPC (automotive, avionics)
  - Complete work on L3 port to ACL2
- **Up-to-Date Executable Formal Models for Common VMs**
  - JVM (invokedynamic), LLVM (a highly moving target)
- **Basic ACL2 -> VM -> Machine Code Verified Compiler**
  - Inspiration: CakeML (verified HOL4 -> ML -> machine code)
  - Current Verified compilers don’t generate LLVM or other SSA Form
- **Verified Simple REPL with Verified GC, Verified Bignums**
  - Reuse CakeML Runtime?
- **Capable Verification Environment for VMs and Machine Code**
  - Codewalker
  - Low-level equivalence checking (Axe, AIGs)

Some Wilder and Crazier Ideas

- **Use Refinement-Based Techniques (Kestrel) for Machine Models**
  - Arbitrary-Precision LLVM to 64-bit LLVM
  - Infinite Memory Size to Finite Memory Size
- **Run Verified Machine Code on Verified Machine Model**
  - CakeML REPL running on UT x86-64 model
  - seL4 running on ACL2 version of ARM model
- **Failed Inductive Subgoal Advisor in Proof Checker**
  - Ex: Identify “key hypothesis”; suggest sequence of rewrites to make the “key hypothesis” equal to the conclusion
- **Use Machine Learning Techniques to Discover Theorems Involving n ACL2 Primitives in the Background**
  - “Discover theorems involving take, nth, nthcdr, and update-nth”