ACL2-2014
Panel Session

David Hardin

Machine Code Verification: Past, Present, and Future

Overview

- ☐ What was feasible in machine code verification 10 years ago?
- ☐ What is feasible now?
- ☐ What will be feasible 10 years from now?

Past: AAMP7 (2003)

- ☐ Verified that the partitioning microcode of the Rockwell Collins AAMP7 upheld a high-level separation theorem (GWV)
- ☐ ACL2 proofs utilized as certification evidence in important Rockwell Collins information assurance products
- ☐ ACL2 was the sole proof tool used
- ☐ 300 lines of microcode
- ☐ Required much low-level memory modeling (reusable)

Present: seL4

- 8000 source line kernel
- Functional verification to C source level carried out using Isabelle/HOL
- Verification to ARM assembly code level accomplished by Magnus Myreen using his "decompilation into logic" approach in HOL4
  - Myreen used a SAT solver to assist with his proofs

Future

- ☐ Wirth: Algorithms + Data Structures = Programs
- ☐ Can readily translate imperative programs to functional form (Myreen's decompilation into logic)
- ☐ Many classic algorithms on algebraic data types can be proved to uphold key properties, esp. with the aid of modern solvers (see Rada)
- ☐ Data structures used by machine code programs are often inconvenient for analysis (lists embedded in arrays, etc.)
- ☐ Need to formally abstract these data structures, then do proofs