3.33 Finding and Responding to Failure in Large Formal Verifications

Mark Staples (NICTA, AU)

The L4.verified project has completed the formal verification, to the level of C source code, of the full functional correctness of the seL4 microkernel [2]. The project proceeded in several phases, with internal iterations, and ongoing maintenance [1]. The team can now claim there are zero bugs in the microkernel, subject to assumptions and conditions. The scale and detail in the project raise challenges of potential interest for Artificial Intelligence (AI) and Formal Methods (FM). I discuss two of these.

Firstly, changes to the specification, design, code, and invariants are almost inevitable in large formal verification projects, because of bug fixes or enhancements. Changes mean the system must be re-verified, and all lemmas must be re-proved. The automated re-proof of some lemmas may fail, because they are no longer true and must be reworked, or because the proof scripts are too fragile and must be reworked. Reverification is a management and technical challenge for which AI techniques may be relevant. Specific challenges include: In what order should lemmas be reworked? How can proof scripts be made more robust to minor changes to lemmas?

Secondly, the existence of extra-logical "gaps" between formal models and the real world (actual requirements and implementations) is well known in the FM community. Formal methods are empirical too, because the properties proved about software are claims about how it will behave and satisfy requirements in the world. Nonetheless, it is less well known how to address these gaps. The L4.verified team identified some extra-logical assumptions, including that assembly-code, C compiler, and hardware are correct. What other key assumptions might there be, and how can we identify them? Can heuristic search techniques help to identify or test such assumptions?

References

1 J. Andronick, D. R. Jeffery, G. Klein, R. Kolanski, M. Staples, H. Zhang, and L. Zhu. Large-scale formal verification in practice: A process perspective. In M. Glinz, G. C. Murphy, and M. Pezzè, editors, *34th International Conference on Software Engineering, (ICSE 2012)*, pages 1002–1011. IEEE, 2012.

2 G. Klein, J. Andronick, K. Elphinstone, G. Heiser, D. Cock, P. Derrin, D. Elkaduwe, K. Engelhardt, R. Kolanski, M. Norrish, T. Sewell, H. Tuch, and S. Winwood. seL4: formal verification of an operating-system kernel. *Commun. ACM*, 53(6):107–115, 2010.