Scott Buckley – Formal Methods and Verification

scott@buck.ly, https://scott.buck.ly, +61431663164

I am a formal methods and verification expert with experience formally specifying and completing mechanised proofs about complex systems, as well as writing fast verification tools. I completed my PhD specifying the operational semantics for dynamically-scheduled attribute grammar evaluation, and mechanising proofs about these semantics. I then spent four years specifying and verifying the prevention of microarchitectural timing channels in seL4, and in 2024 I built a program synthesis tool in C++ using a new safety-game based LTL synthesis algorithm.

Education

Ph.D, Computer Science
Macquarie University, 2015 - 2019. Advisor: Anthony Sloane.
Thesis title: Foundational Semantics of Dynamically-Scheduled Attribute Grammar Evaluation.

Masters by Research, Computer Science
Macquarie University, 2013 - 2014. Advisor: Matthew Roberts.
Thesis title: Attribute Grammars: An Executable Specification for CSS Layout.

Bachelor of Advanced Science in Software Technology
Macquarie University, 2008 - 2012

Experience

Research Associate (Postdoctoral Fellow)
University of New South Wales, Jan 2025 - present. Advisor: Ron Van der Meyden
Here I am working on the MCK (Model Checker for Knowledge) project, an epistemic model checker. The goal of this project is to build a smart contract verification tool that leverages MCK's ability to reason about epistemic properties, to allow for properties to checked on Ethereum smart contracts that could not be specified or checked using existing tools. Specifically we are aiming to model the flow of knowledge through transaction requests visible in public mempools, and how this information leak can lead to otherwise-undetectable security vulnerabilities.

Research Associate (Postdoctoral Fellow)
Université Libre de Bruxelles, Jan 2024 to present. Advisors: Jean-François Raskin, Emmanuel Filiot.
Postdoc in game theory and automata, working on LTL synthesis using a new active learning approach. The focus here is in phrasing sub-problems as safety games (which can be solved quickly) and inclusion queries against a program specification to learn a minimal strategy that satisfies the specification. My main task was to develop a proof-of-concept tool to demonstrate the applicability of this approach, which was completed quickly, and the role evolved into a focus on refining the tool to compete in future synthesis competitions.

A fun part of this role has been to write a sophisticated tool from scratch for an abstract verification problem in C++, focusing both on high-level graph theory concepts alongside paying attention to low-level programming concerns, making the tool as fast as possible.

Research Associate (Postdoctoral Fellow)
University of New South Wales, Jan 2020 - August 2023. Advisors: Toby Murray, Gernot Heiser, Gerwin Klein.

Formal methods and verification postdoc, extending the seL4 proof base with microarchitectural and fine-grained timing semantics, to prove the absence of timing side-channels in seL4, using Isabelle.

In this role I developed a model of a generic separation kernel with underspecified microarchitectural behaviours, such that information-flows through microarchitecture are present. Extending the model with "time protection" mechanisms (spatial and temporal microarchitecture partitioning) allowed for successful proofs of non-interference.

I also developed and implemented a strategy to integrate time protection proofs with the existing seL4 proof-base (1M+ SLOC), which required extracting generalised hardware-interaction traces from seL4's abstract specification. The research and design work for this project was completed, although the proof effort required more engineering time to complete at the time of my departure. This specification and its proofs have already revealed errors in the existing time protection mechanisms in seL4.

Research Programmer
Macquarie University, Apr - Dec 2019
Built and modified simple compilers in Scala, implementing Language Server Protocol libraries for various toy languages in Kiama, as well as the Kiama language library itself.

Foundation Teacher (Casual Lecturer)
Macquarie University International College, Jul - Sept 2018
Preparing and lecturing both lectures and tutorials for an introduction to programming unit. I turned down an offer to continue lecturing due to PhD commitments.

Casual Academic Tutor
Macquarie University, 2013 - 2019 (and Aug - Nov 2023)
Preparing and presenting tutorial workshops for 7 different units in first-year to third-year computer science. Programming, algorithms, OOP, PL, computer architecture, web tech.

Open Source Contributions

- **hevm**: "hevm is an implementation of the Ethereum virtual machine (EVM) made for symbolic execution, equivalence checking, and unit testing of smart contracts." (Ethereum Foundation) https://github.com/ethereum/hevm

Publications

- Scott Buckley, Robert Sison, Toby Murray, Gerwin Klein, and Gernot Heiser (2023). "Proving the Absence of Microarchitectural Timing Channels". Preprint. http://arxiv.org/abs/2310.17046v1
- Robert Sison, Scott Buckley, Toby Murray, Gerwin Klein and Gernot Heiser (2023). "Formalising the Prevention of Microarchitectural Timing Channels by Operating Systems". In: *Proceedings of the 25th International Symposium on Formal Methods*. pp. 103-121.
- Buckley, Scott and Anthony M Sloane (2017). "A formalisation of parameterised reference attribute grammars". In: *Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering*. ACM, pp. 139–150.
- Buckley, Scott, Anthony Sloane, and Matthew Roberts (2016). "Specifying CSS layout with reference attribute grammars". In: *Companion Proceedings of the 2016 ACM SIGPLAN International Conference on Systems, Programming, Languages and Applications: Software for Humanity*. ACM, pp. 29–30.
- Sloane, Anthony M, Franck Cassez, and Scott Buckley (2016). "The sbt-rats parser generator plugin for Scala (tool paper)". In: *Proceedings of the 2016 7th ACM SIGPLAN Symposium on Scala*. ACM, pp. 110–113.

- Sloane, Anthony M, Matthew Roberts, et al. (2014). “Monto: A disintegrated development environment”. In: *International Conference on Software Language Engineering*. Springer, Cham, pp. 211–220.

Other Experience

- Experience in programming languages and ITPs: Isabelle, Coq, Lean, JavaScript, Scala, Java, Python, RISC assembly, Processing, C++, C, PHP, SQL, Arduino.
- Some personal projects:
  - Sudoku (and variant) puzzle solving interface and automatic solver, which can solve extremely difficult puzzles *with no guessing or backtracking*. https://scott.buck.ly/Sudoku.
  - JavaScript-based solvers for other puzzle types: Masyu, Slitherlink, StarBattle etc.
  - Browser-based music player and library management platform, complete with MP3 parsing, implemented (insanely) as a single 3.5k line PHP file. No longer hosted publicly.
  - Interactive SPARC Emulator in JavaScript (for an undergraduate research project in 2012).
- Deep personal interest in logical puzzles – a personal highlight: being the second person world-wide to solve an extremely difficult Sudoku variant, before any of the CTC testers were able to solve it. https://youtu.be/K3O-jvc4HOs.
- Published various 3D-printable designs and mods for the Ender 3 V2 3D printer, used and remixed many times by the community https://www.thingiverse.com/mrwraith2/designs.

Personal Miscellany

- A total of ten weeks spent volunteering at various animal conservation projects in Namibia and Thailand.
- Poured beers as a volunteer at eight different beer festivals in Belgium, The Netherlands, Japan, Denmark, England, and Australia.
- Assisted as a temporary brew hand at six different craft breweries in The Netherlands and Belgium, including six months at Oedipus Brewing in Amsterdam, NL.

Referees

Referees available on request.