Edward Pierzchalski

Email: edward@pierzchalski.org Mobile: +614 04 664 375 Github: pierzchalski

Education

University of New South Wales

- Bachelor of Science (Mathematics), graduated February 2016
- Bachelor of Science (Computer Science)
- Pursuing Honors in Computer Science, expected graduation February 2017

Awards

- Google Third Year Prize
- CSE Undergraduate Performance Prize Year 3, 4th Place

Notable Work

Advanced Algorithmic Verification: Gained experience using Isabelle, an interactive theorem prover. Formally verified a simple C program. Ranked 1st in the course.

Concepts of Programming Languages: Implemented a small Haskell variant, including typechecker. Ranked 1st in the course.

Advanced Operating Systems: Wrote a kernel on top of the SeL4 microkernel. Written in Rust, a new systems programming language.

Computer Graphics: Produced an interactive graphical demo including modern shaders. Achieved a High Distinction.

Systems Capacity Planning: Applied Monte-Carlo simulation techniques to model a constrained-capacity network. Achieved a High Distinction.

Publications

Franck Cassez et al. "Perentie: Modular Trace Refinement and Selective Value Tracking". In: *Tools and Algorithms for the Construction and Analysis of Systems*. 2015, pp. 439–442. URL: http://dx.doi.org/10.1007/978-3-662-46681-0_39.

Toby Murray et al. "Compositional Verification and Refinement of Concurrent Value-Dependent Noninterference". In: *IEEE Computer Security Foundations Symposium*. Lisbon, Portugal, June 2016.

Employment History

Data61

Research assistant, July 2014 to November 2015, February 2016 to present

- Independently investigating formal semantics of concurrent security-sensitive programs, with a focus on value-dependent classification.
- Developed judgement systems to help automate analysis of security properties in a simple imperative language.
- Verified properties of judgement system using Isabelle, a theorem prover.

Google Australia

Engineering intern, November 2015 to February 2016

- Extended collection of public user G+ post data, applied indexing and text salience techniques to extract popular content. Written in Go.
- Demonstrated ability to communicate and coordinate with offsite teams, coding and execution skills, and ability to familiarize myself with internal tools.

NICTA

Software engineering intern, March 2013 to July 2014

- Engineer on the *Perentie* team, an experimental software verification tool written in Scala. Participated in SV COMP 2015, a software verification competition. *Perentie* ranked third in its category after less than a year of development.
- Designed intermediate languages to represent and simplify C program semantics.
- Helped design and implement an embedded domain-specific language for interacting with SMT solvers.

Skills

- Experienced in research and software development, independently or in small to medium teams.
- Strong background on functional programming, language design and semantics, and type theory.
- Proficient in Rust, C, Haskell, Python, LaTeX, and Scala.