Colm Baston

| Email: | colm@colmbaston.uk | Web: | colmbaston.uk |
| Mobile: | +44 (0)7712212743 | Projects: | gitlab.com/colmbaston |

Profile

Recently completed a PhD in Computer Science at the Functional Programming Lab, University of Nottingham, due to graduate in July 2025. I am interested in functional programming, type systems, and formal methods which provide strong, static guarantees about program behaviour. I am excited to apply my knowledge of computer science theory to develop clear, correct, and efficient software to solve real-world problems.

Education

Oct 2016 – Present

Functional Programming Lab, the University of Nottingham
PhD in Computer Science
– Thesis titled *Agent-Based Logics in Dependent Type Theory*, submitted in September 2024, viva examination passed in January 2025, graduation in July 2025, with supervision from Venanzio Capretta and Graham Hutton.
– Investigated the use of dependently-typed programming languages, serving as proof systems, to model the semantics of agent-based logical systems.
– Presented peer-reviewed academic work at two international conferences, with publications and other associated files available at colmbaston.uk.

Sept 2012 – June 2016

School of Computer Science, the University of Nottingham
MSci (Hons) in Computer Science
– Graduated with First Class Honours in July 2016.
– Dissertation titled *Fundamentalist Functional Programming*, developing a command-line interpreter for the untyped λ-calculus using Haskell.
– Guru in the *Computer Science Guru Programme* for two years, a student volunteer aiding first-year students integration into university life and studies.
– Member of group awarded the *School Prize for Best Group Project, 2013-14*.

Experience

Feb 2015 – June 2023

School of Computer Science, the University of Nottingham
Teaching Assistant
– Provided teaching support for 10+ undergraduate modules offered by the school, interacting with hundreds of undergraduates each year.
– Modules include *Mathematics for Computer Scientists*: set theory, predicate logic, formal proofs, combinatorics, probability theory, graph theory;
– *Algorithms, Correctness, and Efficiency*: algorithms, data structures, proving their correctness and complexity;
– *Programming Paradigms*: purely-functional programming with Haskell;
– *Advanced Functional Programming*: further functional programming with Haskell, emphasising practical programming with IO, state, and exceptions;
– *Compilers*: language syntax and semantics, parsing, type-checking, code generation, developing a compiler for a simple imperative language using Haskell.

June 2013 – Sept 2013

Remsdaq Ltd, Deeside, Flintshire
IT Assistant
– Managed the transition of IT infrastructure to a virtualised server environment, provided technical support for all of the other departments on-site.

Skills

– Able to read and write code in a range of programming languages across multiple paradigms, preferring languages with strong, static type systems, most often choosing Haskell (selected projects on GitLab: lambda-interpreter, mtc, statusbar, polytile) and Rust (advent-of-code, festive-bot, kerbolar, badger).

– Strong computer science theory background, with interest in mathematical logic, including formal proof experience using systems like Coq and Agda. Knowledge of algorithms, data structures, and their analysis, able to reason about their correctness and efficiency.

– Proficient with version control systems, particularly Git, and most comfortable when working in Unix-like environments and with the utilities that accompany them, all personal systems running Arch Linux.