

Education

September 2008-December 2012 **PhD in Computer Science, University of Cambridge.**

Dissertation: "Supercompilation for call-by-need languages"

September 2005-June 2008 **BA (first class) in Computer Science.**

Dissertation: "SQL-style list comprehensions for the Glasgow Haskell Compiler"

Awards: 2 AT&T Prizes (for the best first class degrees in the second and third years), Microsoft Research Prize (for the best third year dissertation), Victor Chu Prize For Academic Achievement (for best performer in the college), 3 College Prizes, College Scholarship, Senior College Scholarship

September 2003-June 2005 **A levels (AAAAA), The Henley College.**

Computing, Economics, Electronics, Mathematics and Physics

Awards: Best overall A-level student (of a year of approximately 1000 students)

2002 **Microsoft Certified Solution Developer.** Achieved by independent study.

Work

July 2009-November 2009 **Quantitative developer, Standard Chartered.**

Developing technology for product valuation using Haskell, Java and C++.

Worked as part of the core technology team building a functional-programming based platform used by quants and traders for valuation. Principally involved in improving the build system for the platform, building a compiler for an custom programming language and developing two tools for programmers to used to write SSE-based inner loops: a C++ template library, and a Haskell combinator library based on "causal commutative arrows".

June 2008-August 2008 **Summer of Code developer, Google.**

Accepted to Google's annual open source contribution program.

Remotely collaborated with mentors in Cambridge and Australia to design, implement and document plugins for the Glasgow Haskell Compiler. Compiler plugins allow the user of GHC to dynamically link their own optimising transformations or program invariant checks into the compiler itself. As well as implementing dynamic linking of passes, a phase control system and annotation system were added to Haskell. Design rather than implementation work was a large portion of this project, as there was a large design space without clear best choices.

June 2007-October 2007 **Front office developer, KBC Financial Products.**

Working solo and as part of a team to meet front office needs with Java and C#.

Worked on fundamental Java infrastructure designed to deliver news stories from Reuters and Bloomberg to both traders and news arbitrage based algorithmic trading strategies. This was completed ahead of schedule despite the inadequate documentation and code libraries available from the news providers. Later contributed to a C# application for structuring CDO² products. Implemented reliability and code quality improvements, as well as tactical feature development in response to the needs of the front office. My internship was extended for 3 weeks in light of the quality of my contributions.

June 2006-October 2006 **Software developer, Resolver Systems.**

Full member of team building Python-based spreadsheet with C# and IronPython.

Resolver practiced “extreme programming”, so all development was test-driven and used pair-programming. Implemented and tested multiple spreadsheet features using IronPython, a CLR-based Python variant. I made particular contributions towards build tools: maintaining a continuous integration server, and writing a distributed build system.

June 2004-September 2004 **Software developer, Research Machines.**

Working solo to automate internal processes with Access VBA and C#.

Developed an accounting system for departmental cross-charging which interfaced with the existing web of departmental databases and spreadsheets that were used to manage people and projects. After a phase of requirements capture which involved consulting across the department I implemented and deployed a solution using VBA in half the time that had been allocated for the task. Went on to develop a tool to transform incoming XML files containing examination data into the poorly specified binary format of another legacy system: the implementation technology was C#. This involved working with the American vendor of the legacy system to ensure that my tool remained compatible with it.

June 2003-September 2003 **Software tester, Research Machines.**

Executing and writing test scripts for an educational portal, Visual Basic development.

Worked as part of a large team of testers finding bugs. Recorded video tutorials for the portal. Wrote a web-based ASP timesheet application for tracking hours worked by permanent and temporary employees internationally. This involved significant integration work with the existing departmental accounting infrastructure.

Publications

June 2011 **“Termination combinators forever”** with Simon Peyton Jones and Dimitrios Vytiniotis. Presented at the Haskell Symposium ‘11. *Describes a representation of well-quasi-order theory as a combinator library.*

July 2010 **“Supercompilation by evaluation”** with Simon Peyton Jones. Presented at the Haskell Symposium ‘10. *Powerful automatic program optimisation for call-by-need functional languages.*

May 2009 **“Ypnos: declarative, parallel structured grid programming”** with Dominic Orchard and Alan Mycroft. Presented at Declarative Aspects of Multicore Programming ‘10. *A language for distributed computations (such as computational fluid dynamics) with spatially static data dependencies.*

May 2009 **“Types are calling conventions”** with Simon Peyton Jones. Presented at the Haskell Symposium ‘09. *A compiler intermediate language with explicit arity information.*

Points of interest

- Developed the most popular testing framework for Haskell, supporting full parallel test execution without non-deterministic output: <http://batterseapower.github.com/test-framework/>
- Built a custom alias analysis for LLVM: <http://blog.omega-prime.co.uk/?p=135>
- Implemented an advanced type system feature for Haskell, “constraint kinds”, allowing abstraction over type classes: <http://blog.omega-prime.co.uk/?p=127>
- Built a Chinese learning tool used by thousands of people: <http://batterseapower.github.com/pinyin-toolkit/>
- Designed and implemented Unicode support on the command line for Haskell (PEP-383)
- Experience reverse-engineering file formats and network protocols
- Technical reviewer for the books “IronPython in Action” and “Ruby in Practice”