Curriculum Vitae for Professor Neil Ghani

Personal Details:

Name:	Prof. Neil Ghani	**D.O.B.:**	21 July 1967
Address:	Dept. of Comp. and Inf. Sci., University of Strathclyde, Livingstone Tower, Glasgow, Scotland	**Nationality:**	British
		Email:	ng@cis.strath.ac.uk

Academic Qualifications:

October 1990 to June 1995	*Adjoint Rewriting* Ph.D. in Computer Science, supervised by Prof Don Sannella L.F.C.S., Dept. of Comp. Sci., Univ. of Edinburgh.
October 1989 to June 1990	M.Sc. in Computation. Balliol College, Oxford University, Oxford.
October 1985 to June 1988	B.A. Hons 2:1 in Mathematics. Pembroke College, Oxford University, Oxford.

Career History:

November 2014 —	Associate Dean Research, Faculty of Science, Univ. of Strathclyde, Glasgow, Scotland.
July 2008 —	Prof. of Computer Science, Univ. of Strathclyde, Glasgow, Scotland.
September 2005 to July 2008	Lecturer and Reader, School. of Comp. Sci. and IT Univ. of Nottingham, Nottingham, England.
September 1998 to August 2005	Lecturer, Dept. of Comp. Sci. Univ. of Leicester, Leicester, England.
January 1997 to August 1998	Research Fellow, Univ. of Birmingham, Birmingham B15 2TT, England.
December 1995 to Dec. 1996	Research Fellow, Ecole Normale Superieure, 45, Rue d'Ulm, 75230 Paris Cedex 05, France.

Honours and Distinctions:

- Election to EPSRC Peer Review College, 2006-2010 and 2014-
- Election to steering committee of BCTCS, 2004-2007.
- Election as Director of the Midlands Graduate School (MGS), 2003-2006.
- Scholarship to study Mathematics at University of Oxford, 1985-1988.

Selected Grant Income

- Homotopy Type Theory: Programming and Verification. EPSRC 2015-2019. Principal Investigator. Grant Value: 1,002,000 pounds.
- Logical Relations for Program Verification. EPSRC 2013-2017. Principal Investigator. Grant Value: 442,000 pounds.
- Correct By Construction Mathematics. EU IRSES Project 2014-2018. Co-Investigator. Grant Value: 250,000 Euro spread across 15 sites.
- Implementing Units of Measure. Impact Accelerator Grant. 2014-2015. Principal Investigator. Grant Value 10,000 pounds.
- Categorical Foundations of Indexed Programming. EPSRC 2010-2013. Co-Investigator. Grant Value: 282,000 pounds.
- Reusability and Dependent Types. EPSRC 2009-2012. Principal Investigator. Grant Value: 517,000 pounds.
- Theory and Applications of Induction Recursion. EPSRC 2009-2013. Principal Investigator. Grant Value: 523,000 pounds.
- Theory and Applications of Containers. EPSRC 2005-2008. Principal Investigator. Grant Value: 229,000 pounds.
- Midlands Graduate School. EPSRC 2004-2006. Principal Investigator. Grant Value: 12,000 pounds.
- Applied Semantics II. European Union. Local contact for University of Leicester. Grant Value: 3,000 pounds.
- BCTCS 2003. The London Mathematical Society, 2003. Sole Investigator. Grant Value: 4,000 pounds.
- Coalgebra and Recursion. The Royal Society of London, 2003-2005. Sole Investigator. Grant Value: 6,000 pounds.
- Kan - A Categorical approach to Computer Algebra. EPSRC 2001-2004. Sole Investigator. Grant Value: 130,000 pounds.
- Categorical Rewriting: Monads and Modularity. EPSRC 2000-2002. Sole Investigator. Grant Value: 52,000 pounds.
- Eta-Expansions in Dependent Type Theory. EUROFOCS 1996. Sole Investigator. Grant Value: 10,000 pounds.
- Eta-Expansions in Dependent Type Theory. The Royal Society of London 1996. Sole Investigator. Grant Value: 2,500 pounds.

Research Interests and Achievements:

I am trying to understand the structure of computation and to turn that understanding into the next generation of programming languages. I see the mathematical foundations of computation and programming as inextricably linked. We study one so as to develop the other. This reflects the symbiotic relationship between mathematics, programming, and the design of programming languages any attempt to sever this connection will diminish each component. This is quite a bold research program and I realise that, inevitably, only partial answers will be forthcoming. Nevertheless, it shows my commitment to ask deep and fundamental questions so as to produce research that is of the highest calibre, and that will stand the test of time rather than becoming obsolete within a few years. Indeed, the simplicity afforded by such deep insights seems to me to be the only way to develop theories that can be scaled successfully to more applied areas such as software development. The ultimate practical end point of my desire to understand the nature of computation is to develop program language and program verification tools - an area of clear importance as recognised by both EPSRC and the EU.

In my research, I typically use categorical methods as a semantic description of computation, type theory as an intermediate language between this semantics and actual programming languages, and functional programming as a target model of computation. As a result, my research is usually innovative in making hitherto unseen connections between these disciplines and more mainstream computer science areas such as rewriting, computational algebra, and artificial intelligence. I regard my role as not simply to prove results, nor just to write papers, but to build a substantial body of evidence to substantiate my research agenda and to provide the leadership and inspiration that will allow others to join in my vision. I honestly believe that this is an extraordinarily exciting time to be a computer scientist - we face great challenges but also have some amazingly powerful tools at our disposal. The following is a brief summary of my research to date:

- **Category Theory:** Category theory is a relatively new mathematical discipline that provides an abstract theory of structure, and hence is key to my work. Firstly, I showed how various computationally interesting structures such as rewrite systems, term graphs, infinitary terms and cyclic data structures are monadic in nature. Secondly, while monads model the description of computational agents, their evolution is modelled as coalgebras. I helped describe the relationship between coalgebras and monads, and gave a coalgebraic foundation to various sophisticated bisimulations in the π-calculus. A grant from the Royal Society of London has funded part of this research. Recently, I have been using category theory to develop a uniform model of indexed computation, and this research has been funded by EPSRC with a grant of 282,000 pounds. Most recently, I have been awarded a new grant from EPSRC worth 440,000 to develop a new understanding of logical relations.
- **Rewriting:** Rewriting Systems are widely used as abstract models of computation, since they are computationally expressive even while retaining a relatively simple and concrete syntax. My thesis developed the subject of η-expansions and showed it to be better behaved than the more traditional theory of eta-contractions. I solved the long standing open problem of the decidability of $\beta\eta$-equality for sum types, which had attracted the attention of a number of research groups across the world. I made the connection between rewrite systems and universal algebra via monads and deduced new results in modular rewriting (see the section on artificial intelligence). I had an EPSRC grant for 52,000

pounds to conduct research in this area and a special workshop of Rewriting Techniques and Applications was devoted to my results. This research was classified as outstanding by the EPSRC panel which considered the associated IGR reports.

- **λ-Calculus:** The λ-calculus is of foundational importance within computer science and, in particular, can be viewed as a paradigmatic functional programming language. I have worked on features such as type systems, pattern matching, and explicit substitutions, which make the lambda-calculus closer to "real" functional languages. I was awarded grants from the Royal Society of London and from the EUROFOCS programme to conduct part of this research.

- **Functional Programming:** Monads are a useful abstraction of computation as they model diverse computational effects such as stateful computations, exceptions, and I/O all in a uniform manner. I developed a new approach to monad composition which is i) general in that nearly all monads compose; ii) mathematically elegant in using the standard categorical tools underpinning monads; and iii) computationally expressive in supporting canonical recursion operators.

  My research on containers has provided new and surprisingly simple results on the nature of polymorphism for concrete data types, shown how coinductive types can be constructed from inductive types, and shown how data types support a generic notion of differentiation. I am currently developing what we believe will become the next generation of data types by using containers to develop a grammar for inductive families together with an implementation of this grammar in Epigram. I was awarded an EPSRC grant for 230,000 pounds to further this research. I'm also working on induction recursion which I believe will become the generation-after-next's theory of data types, and this research has been awarded a grant of 311,000 pounds by EPSRC.

  Initial algebra semantics is the corner stone of inductive types, but I have shown that the theory as usually presented is incomplete. In particular, I extended initial algebras with an additional universal property based upon the characterisation of initial algebras as limits. This has generated new Church encodings for inductive types, placed short cut fusion at the centre of initial algebra semantics, and given the first generalisation of these concepts to nested types and GADTs. I am also working on the development of an initial algebra semantics for GADTs so as to extend techniques developed for nested types to GADTs.

  Finally, I have become very interested in the use of dependent types to model semantic information within programs themselves, and this research has been funded by an EPSRC grant with a 110,000 pounds. Within this project, I am interested in how one may built new data types from old data types in a programming language. Currently, one has to simply build all new data types from scratch - this is both wasteful and inefficient.

- **Computational Algebra:** Computer algebra packages are widely used in mathematics and computer science to solve combinatorial problems whose essence is the computation of the quotient of an algebraic structure. Current packages tend to be simply a collection of disparate algorithms derived on an ad-hoc and case by case basis. However, I observed that most quotients are examples of the computation of a left Kan extension including cosets, double cosets, orbits and modules etc. EPSRC funded a project to develop generic algorithms for computing quotients with a grant of 130,000.

- **Artificial Intelligence:** When reasoning about complex systems (such as specifications of large systems, or semantics of rich languages with many different

features), modularity is a crucial property. It allows properties about smaller (and hence easier to reason about) components to be lifted to the overall system. As with other problems mentioned above, I want to understand the essence of modularity so as to develop abstract methodologies which apply to a variety of different modularity problems where more concrete approaches fail to deliver. To this end, I developed a new semantic framework for modularity based upon monad combinators, developed the algorithms required to implement this framework and applied these theoretical results to a number of specific problems.

To summarise, I've developed new decision procedures in rewriting, new theories of data types in functional programming, new algorithms in computational algebra, and new modularity results in artificial intelligence. I've published them in the leading international conferences in those application areas so as to disseminate those ideas widely and to learn from the feedback of these communities. Crucially, the key to these results has been to first understand the deep structure of these problems. This interaction between practical problems and theoretical insight seems to me to be the hallmark of good science.

Teaching

I am a passionate teacher who motivates students to succeed in their studies by sharing with them my own enthusiasm for learning new ideas. I aim to demonstrate to students how the material being taught is chosen for a reason — it allows us to do things more simply than would otherwise be the case. This is important because, I believe, students want to learn but can lose this desire if material is poorly presented. Overall, I show students how education enables them to act more effectively within the world and thereby enhance and enrich their lives. I also believe that learning, like life, is an active process, and so involve students as much as possible as active participants in lectures. In particular, I ensure all students participate, not just those with outgoing characters or those who understand the material being taught.

I have taught at the Universities of Leicester, Nottingham, and Strathclyde and also given regular postgraduate courses at the MGS (see Academic Service section) and taught at the Estonian Winter School in 2003. In each of these venues, colleagues and students report that my teaching style is very effective. While at Leicester, I was regarded as one of our best teachers. This was demonstrated through both the formal teaching evaluation mechanism and also through informal feedback. One of the courses I taught at Nottingham was rated the second most popular in the school by the students. At Strathclyde, the feedback from students and colleagues continues to be equally positive. I have taught the following courses:

C01003	Yr 1, University of Leicester	Program Design
C01004	Yr 1, University of Leicester	Algorithms and Data Structures
C02008	Yr 2, University of Leicester	Functional Programming
C02015	Yr 2, University of Leicester	Group Project
C03012	Yr 3, University of Leicester	Individual Dissertation
C03097	Yr 3, University of Leicester	Programming Secure and Distributed Systems
G51FUN	Yr 1, University of Nottingham	Functional Programming
G51PRG	Yr 1, University of Nottingham	Programming in Java
G52GRP	Yr 2, University of Nottingham	Group Project

G5BIAW	Yr 3, University of Nottingham	Internet and the WWW
G53IDS	Yr 3, University of Nottingham	Individual Dissertation
52.231	Yr 2, University of Strathclyde	Programming Techniques
52.222	Yr 2, University of Strathclyde	Programming Project
CS203	Yr 2, University of Strathclyde	Topics in Computer Science
CS208	Yr 2, University of Strathclyde	Algorithms and Data Structures
CS316	Yr 3, University of Strathclyde	Functional Programming

Research Publications

The culture of Computer Science is to both ensure and regard conference publication as being of high quality. For example, all of the conferences I submit to are internationally leading and most reject about 4 papers for every paper accepted.

Journal Papers

- Positive Inductive-Recursive Definitions. Neil Ghani, Lorenzo Malatesta, and Fredrik Forsberg. *Logical Methods in Computer Science, 11(1)*, 2015.
- Containers, Monads and Induction Recursion. Neil Ghani and Peter Hancock. *Journal of Mathematical Structures in Computer Science* 2014.
- Indexed Induction and Coinduction, Fibrationally. Clement Fumex, Neil Ghani, and Patricia Johann. *Logical Methods in Computer Science, 9 (3:6)*, 2013.
- Refining Inductive Types. Robert Atkey, Patricia Johann, and Neil Ghani. *Logical Methods in Computer Science 8(2)*, 2012.
- Generic Fibrational Induction. Neil Ghani, Patricia Johann, and Clement Fumex. *Logical Methods in Computer Science* 8(2), 2012
- A principled approach to programming with nested types in Haskell. Patricia Johann and Neil Ghani. *Journal of Higher-Order and Symbolic Computation*, Volume 22(2), pages 155-189, 2010.
- Representations of Stream Processors Using Nested Fixed Points. Neil Ghani, Peter Hancock and Dirk Pattinson. *Logical Methods in Computer Science*, Volume 5(3), 2009.
- A Universe of Strictly Positive Families. Neil Ghani, Peter Morris and Thorsten Altenkirch. *International Journal of Foundations of Computer Science*, Volume 20(1), pages 83-107, 2009.
- Monadic Augment and Generalised Short Cut Fusion. Neil Ghani and Patricia Johann. In *Journal of Functional Programming*, Volume 17(6), pages 731 - 776, 2007.
- Explicit Substitutions. Neil Ghani, Tarmo Uustalu and Makoto Hamana. *Journal of Higher Order Symbolic Computation*, vol. 19(2,3), pages 263-282, 2006.
- Computing with Double Cosets. Neil Ghani, Anne Heyworth, Ronnie Brown and Chris Wensley. *Journal of Symbolic Computation*, Volume 41(5), pages 573-590, 2006.

- δ for Data — Differentiating Data Structures. Neil Ghani, Michael Abbott, Thorsten Altenkirch and Conor McBride. *Fundamentae Informatica*, Volume 65(1,2), pages 1-28, 2005.
- Containers - Constructing Strictly Positive Types. Neil Ghani, Michael Abbott and Thorsten Altenkirch. *Journal of Theoretical Computer Science*, Volume 341(1), pages 3-27, 2005.
- Monads of Coalgebras: Rational Terms and Term Graphs. Neil Ghani, Christoph Luth and Federico De Marchi. *Journal of Mathematical Structures in Computer Science*, Volume 15(3), pages 433-451, 2005.
- Coproducts of Ideal Monads. Neil Ghani and Tarmo Uustalu. *Journal of Theoretical Informatics and Applications*, Volume 38(4), pages 321-342, 2004.
- Rewriting via Coinserters. Neil Ghani and Christoph Luth. *Nordic Journal of Computing*, Volume 10(4), pages 290-312, 2003.
- Solving Algebraic Equations using Coalgebra. Neil Ghani, Christoph Luth and Federico De Marchi. *Journal of Theoretical Informatics and Applications*, Volume 37, pages 301-314, 2003.
- Dualizing Initial Algebras. Neil Ghani, Christoph Luth, Federico De Marchi and John Power. *Journal of Mathematical Structures in Computer Science*, Volume 13(1), pages 349-370, 2003.
- Linear Explicit Substitutions. Neil Ghani, Valeria de Paiva and Eike Ritter. *Journal of the International Group on Programming Languages*, Volume 8(1), pages 7-31, 2000.
- The Virtues of Eta-Expansion. Neil Ghani and Barry Jay. *Journal of Functional Programming*, Volume 5(2), pages 135-154, 1996.

Editing of Conference Proceedings

- Co-Editor of *Coalgebraic Methods in Computer Science 2006*. Neil Ghani and John Power. Electronic Notes in Theoretical Computer Science, Volume 164, 2006.

Internationally Reviewed Conference Papers

- Neil Ghani, Fredrik Forsberg, and Alex Simpson. Comprehensive parametric polymorphism: categorical models and type theory. *FOSSACS* 2016. *Awarded Best Theory Paper at ETAPS 2016*
- Neil Ghani, Fredrik Forsberg, and Federico Orsanigo. Proof Relevant Parametricity. *WadlerFest*, 2016.
- Neil Ghani, Fredrik Forsberg, Tim Revell, Patricia Johann and Federico Orsanigo. Bifibrational functorial semantics of parametric polymorphism. *Mathematical Foundations of Program Semantics*, 2015.
- Neil Ghani, Fredrik Forsberg, Tim Revell, Sam Staton, Robet Atkey and Federico Orsanigo. Models for Polymorphism over Physical Dimensions. *Typed Lambda Calculi and Applications* , 2015

- Neil Ghani, Fredrik Forsberg, and Federico Orsanigo. Parametric Polymorphism — Universally. *WOLLIC*, 2015.
- A Relationally Parametric Model of Dependent Type Theory. Robert Atkey, Neil Ghani, and Patricia Johann. *Principles of Programming Languages*, 2014.
- Positive Induction Recursion. Neil Ghani, Lorenzo Malatesta and Fredrik Nordvall-Forsberg. *Conference on Algebras and Coalgebras*, 2013.
- Fibred Data Types. Neil Ghani, Peter Hancock, Lorenzo Malatesta and Fredrik Nordvall-Forsberg. *Logic in Computer Science*, 2013.
- Small Induction Recursion. Neil Ghani, Peter Hancock, Conor McBride, Thorsten Altenkirch, Lorenzo Malatesta. *Typed Lambda Calculus and Applications*, 2013.
- Fibrational Induction Meets Effects. Robert Atkey, Neil Ghani, Bart Jacobs, and Patricia Johann. *Foundations of Software Systems and Computation Structures*, pp. 4257, 2012.
- Indexed Induction and Coinduction in the Fibrational Setting. Clement Fumex, Patricia Johann, Neil Ghani. *Conference on Algebras and Coalgebras*, pp 176191, 2011.
- When Is a Type Refinement an Inductive Type? Robert Atkey, Patricia Johann, Neil Ghani. *Foundations of Software Sciences and Computation Structures*, LNCS 6604, pages 72-87, 2011.
- Modularity and Implementation of Mathematical Operational Semantics. Mauro Jaskelioff, Neil Ghani, Graham Hutton. *Mathematically Structured Functional Programming*, ENTCS Volume 229(5), pages 75-95, 2011.
- Fibrational Induction Rules for Initial Algebras. Neil Ghani, Patricia Johann and Clement Fumex. *Computer Science Logic*, LNCS 6427, pages 336-350, 2010.
- Continuous Functions on Final Coalgebras. Neil Ghani, Peter Hancock, Dirk Pattinson. *Mathematical Foundations of Programming Semantics*, ENTCS Volume 249, pages 3-18, 2009.
- Foundations for Structured Programming with GADTs. Neil Ghani and Patricia Johann. *Principles of Programming Languages*, pages 297-308, 2008.
- Initial Algebra Semantics is Enough! Neil Ghani and Patricia Johann. *Typed Lambda Calculus and Applications*, LNCS 4583, pages 207-222, 2007.
- Constructing Strictly Positive Families. Neil Ghani, Peter Morris and Thorsten Altenkirch. *Australasian Symposium on Theory of Computing*, ACM International Conference Proceeding Series Volume 240, pages 111-121, 2007.
- Continuous Functions on Final Coalgebras. Neil Ghani, Peter Hancock and Dirk Pattinson. *Coalgebraic Methods in Computer Science*, ENTCS Volume 164, pages 141-155, 2006.
- Representing cyclic structures as nested datatypes. Neil Ghani, Makoto Hamana, Tarmo Uustalu and Varmo Vene. *Trends in Functional Programming*, pages 173-188, 2006.
- Monadic augment and generalised short cut fusion. Neil Ghani, Patricia Johann, Tarmo Uustalu and Varmo Vene. *International Conference on Functional Programming*, ACM SIGPLAN Notices Volume 40(9), pages 294-305, 2005.

- Generalising the Augment Combinator. Neil Ghani, Tarmo Uustalu and Varmo Vene. *Trends in Functional Programming*, pages 65-78, 2006.
- Abstract Modularity. Neil Ghani, Michael Abbott and Christoph Lüth. *Rewriting Techniques and Applications*, LNCS 3467, pages 46-60, 2005.
- Relationally Staged Computation in the π-calculus. Neil Ghani, Bjorn Victor and Kidane Yemane. *Coalgebraic Methods in Computer Science*, ENTCS Volume 106, pages 105-120, 2004.
- Constructing Programs with Quotient Types. Neil Ghani, Michael Abbott, Thorsten Altenkirch and Conor McBride. *Mathematics of Programme Construction*, LNCS 3125, pages 2-15, 2004.
- Representing Nested Inductive Types using W-types. Neil Ghani, Michael Abbott and Thorsten Altenkirch. *International Colloquium on Automata, Languages and Programming*, LNCS 3142, pages 59-71, 2004.
- Build, Augment, Destroy. Universally. Neil Ghani, Tarmo Uustalu and Varmo Vene. *Asian Symposium on Programming Languages and Systems*, LNCS 3302, pages 327-341, 2004.
- Difunctorial Semantics of Object Calculus. Neil Ghani and Johan Glimming. *Workshop on Object-Oriented Development*, ENTCS Volume 138(2), pages 79-94, 2005.
- Categories of Containers. Neil Ghani, Michael Abbott and Thorsten Altenkirch. *Foundations of Software Science and Computation Structures*, LNCS 2620, pages 23-38, 2003.
- Derivatives of Containers. Neil Ghani, Michael Abbott, Thorsten Altenkirch and Conor McBride. *Typed Lambda Calculus and Applications*, LNCS 2701, pages 16-30, 2003.
- A Rewriting Alternative to Reidemeister Schreier. Neil Ghani and Anne Heyworth. *Rewriting Techniques and Applications*, LNCS 2706, pages 452-466, 2003.
- Explicit Substitutions and Higher-Order Syntax. Neil Ghani and Tarmo Uustalu. *Mechanized Reasoning about Languages with Variable Binding*, pages 1-8, 2003.
- Computing over K-modules. Neil Ghani and Anne Heyworth. *Computing: The Australasian Theory Symposium*, ENTCS Volume 61, pages 34-50, 2002.
- Monads and Modularity. Neil Ghani and Christoph Lüth. *Frontiers of Combining Systems*, LNAI 2309, pages 18-32, 2002.
- Coalgebraic Monads. Neil Ghani, Christoph Lüth and Federico De Marchi. *Coalgebraic Methods in Computer Science*, ENTCS Volume 65(1), pages 71-91, 2002.
- Composing Monads Using Coproducts. Neil Ghani and Christoph Lu̇th. *International Conference on Functional Programming*, ACM SIGPLAN Notices Volume 37(9), pages 294-305, 2002.
- Algebras, Coalgebras, Monads and Comonads. Neil Ghani, Christoph Lüth, Federico de Marchi and John Power. *Coalgebraic Methods in Computer Science*, ENTCS Volume 44(1), pages 128-145, 2001.
- Categorical Models of Explicit Substitutions. Neil Ghani, Valeria de Paiva and Eike Ritter. *Foundations of Software Science and Computation Structures*, LNCS 1578, pages 197-211, 1999.

- Explicit Substitutions for Constructive Necessity. Neil Ghani, Valeria de Paiva and Eike Ritter. *International Colloquium on Automata, Languages and Programming*, LNCS 1443, pages 743-754, 1998.
- Monads and Modular Term Rewriting. Neil Ghani and Christoph Lüth. *Category Theory and Computer Science*, LNCS 1290, pages 69-86, 1997.
- On Modular Properties of Higher Order Extensional Lambda Calculi. Neil Ghani and Roberto Di Cosmo. *International Colloquium on Automata, Languages and Programming*, LNCS 1256, pages 237-247, 1997.
- Eta-Expansions in Dependent Type Theory — The Calculus of Constructions. Neil Ghani. *Typed Lambda Calculus and Applications*, LNCS 1210, pages 164-180, 1997.
- Eta-Expansions in F^w. Neil Ghani. *Computer Science Logic*, LNCS 1258, pages 182-197, 1996.
- Beta-Eta Equality for Coproducts. Neil Ghani. *Typed Lambda Calculus and Applications*, LNCS 902, pages 171-185, 1995.

Thesis

- Adjoint Rewriting. PhD Thesis, 155 pages, University of Edinburgh, 1995.

Research Supervision

Alasdair Lambert	PhD, University of Strathclyde	since 2016
Ben Price	PhD, University of Strathclyde	since 2015
Tim Revell	PhD, University of Strathclyde	2012-2015
Federico Orsanigo	PhD, University of Strathclyde	2012-2015
Lorenzo Malatesta	PhD, University of Strathclyde	2010-2014
Clement Fumex (2nd)	PhD, University of Strathclyde	2009-2012
Rawle Prince	PhD, University of Nottingham	2006-2009
Mauro Jaskelioff	PhD, University of Nottingham	2005-2009
Michael Abbott	PhD, University of Leicester	2000-2003
Federico de Marchi	PhD, University of Leicester	1999-2002
Fredrik Forsberg	RA, University of Strathclyde	since 2013
Robert Atkey	RA, University of Strathclyde	2010-13
Peter Hancock	RA, University of Strathclyde	2009-2012
Anne Heyworth	RA, University of Leicester	2001-2004

Academic Service

I have demonstrated my ability to discharge senior administrative roles within a University. I have been Associate Dean for Research within the Faculty of Science where I have developed, implemented and monitored policy over the last three years. Further, my Grant Writing Challenge has won wide acclaim with several first-time PIs putting their success significantly down to help from me. I led the *Mathematically Structured Programming* group at the University of Strathclyde from 2008 until 2012.

I was brought in to found the group 5 years ago and it gave me great pleasure to see the immense progress we made. I oversaw the group grow to 13 members with 4 faculty, 2 RAs and 7 PhD students. We have had 8 large grants funded and established an international reputation. I am deputy head of department and participate in the department's annual review process as a reviewer. I was also chair of the research committee of our department and I was our department's representative on our Faculty Research Committee. Finally, I was in charge of preparations for the REF within our department. At Nottingham I was in charge of the organisation of the year long Third Year Projects which all students take. Duties included the allocation of students to supervisors, arranging schedules for vivas and software demonstrations, collation of marks etc. I also went beyond what was expected of me and started the automation of marks management, replacing the sending of marks by email with a web interface for supervisors to enter marks. I was the seminar organiser for my research group within our department and as such invited and looked after our guests. I was on the teaching committee of my department and I was the course director for one of the degrees that we offered. At Leicester I was in charge of all pastoral care within the Department which included the tutorial system, careers provision, matters concerning disability issues, collation of mitigating circumstances for examinations boards etc.

Outside the university, I was the Director of the Midlands Graduate School (MGS) for three years and have thus had overall responsibility for its organisation. This is a collaborative venture between the Universities of Birmingham, Leicester, and Nottingham whose aim is to educate PhD students in state of the art techniques in theoretical computer science which they may not meet in the course of their necessarily highly focused PhD studies. As Director, I obtained the first long term funding for the MGS, supervised the local organisers of the MGS spring schools in 2003, 2004 and 2005, and helped with the selection of lecturing staff and courses.

The British Colloquium on Theoretical Computer Science is a national organisation which provides a forum in which researchers in theoretical computer science can meet, present research findings, and discuss developments in the field. It also provides an environment in which PhD students can gain experience in presenting their work, and benefit from contact with established researchers. I was the local organiser of BCTCS 19, which was held at the University of Leicester in 2003, and for which I obtained funding from the London Mathematical Society. Subsequently, I was elected to the steering committee of BCTCS, which oversees the organisation.

I served on the Engineering and Physical Sciences Research Council (EPSRC) Peer Review College from 2006 until 2010 and have been reappointed to serve a second term. EPSRC is the major government funding body for research in my area and the Peer Review College recommends to EPSRC which grant applications merit funding. This is a significant responsibility to which I was elected by my fellow academics. I have been invited to sit on an EPSRC Grant Allocation panel whose job was to judge the relative quality of research proposals competing for funding. In particular, the panel is responsible for placing the proposals before it in a funding priority order. From this list, the final decision is made on funding. I am also co chair of the SICSA Complex Systems Engineering Theme. Finally I set up and jointly organise the Scottish Category Theory Seminar.

I have given numerous invited seminars in the UK, Europe, and the USA. These include the Universities of Bangor, Bath, Birmingham, Cambridge, Edinburgh, Kent, Nottingham, Oxford, Sussex, Durham, Imperial, Queen Mary College, Bangor, Swansea, Bremen (Germany), Braunschweig (Germany), ENS Paris (France), Milan (Italy), Rutgers (USA), Indiana (USA), and California (San Diego, USA). I have also given keynote

addresses at conferences including MFPS 2009, MSFP 2011, BMC 2013, CALCO 2013, Representing Streams II (2014)

I served on the programme committee of RTA in 2002, 2008, 2009 and CMCS in 2006, 2008, and 2010 as well as PPDP 2008, MFPS 2011, MFPS 2012, FICS 2013, Haskell 2013, and MSFP 2014. I was also the programme co-chair for CMCS 2006 and co-editor of the proceedings of the conference.

References:

Prof Achim Jung	School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT. email: A.Jung@cs.bham.ac.uk tel: 44 121 414 4776
Dr John Power	Department of Computer Science University of Bath, Bath, BA2 7AY email: A.J.Power@bath.ac.uk tel: 44 1225 384439
Prof Martin Hyland	DPMMS, Centre for Mathematical Sciences, University of Cambridge, Cambridge, CB3 0WB email: M.Hyland@dpmms.cam.ac.uk tel: 44 1223 337995