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.

Work Experience:

July 2008 —	Prof. of Computer Science, Univ. of Strathclyde, Glasgow, Scotland.
September 2005 to July 2008	Lecturer, 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, Edgbaston, 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 Engineering and Physical Sciences Research Council (EPSRC - the UK equivalent of National Science Foundation) Peer Review College, 2006-2009.
- Election to steering committee of the British Colloquium on Theoretical Computer Science (BCTCS), 2004-2007.
- Election as Director of the Midlands Graduate School (MGS), 2003-2006.
- Scholarship to study Mathematics at University of Oxford, 1985-1988.

Grant Income

- Theory and Applications of Containers. EPSRC 2005-2008. Principal Investigator. Grant Value: 230,000 pounds. Code EP/C511964/1.
- Midlands Graduate School. EPSRC 2004-2006. Principal Investigator. Grant Value: 12,000 pounds. Code GR/T06087/01.
- Applied Semantics II. European Union. Local contact for University of Leicester. Grant Value: 3,000 pounds. This refers to the funds allocated to Leicester.
- 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. Code GR/R29604/01. End of grant IGR assessment: Tending to Outstanding.
- Categorical Rewriting: Monads and Modularity. EPSRC 2000 - 2002. Sole Investigator. Grant Value: 52,000 pounds. Code GR/M96230/01. End of grant IGR assessment: Outstanding.
- 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. This is quite a bold statement and I realise that, inevitably, only partial answers will be forthcoming. Nevertheless, I make this statement because it shows my commitment to ask deep and fundamental questions so as to produce research which is of the highest calibre and which will stand the test of time rather than become obsolete within a few years. It also brings out the unifying theme present in the different strands of research I have undertaken. In addition to being intersting in its own right, understanding the structure of computation is also a prerequisite for computing with, and reasoning effectively about, advanced computational agents. Indeed, the simplicity afforded by such deep insights seems to me to be the only way to devlop theories that can be scaled succesfully to more applied areas such as software development.

I typically use categorical methods as a semantic description of computation and type theory as an intermediate language between this semantics and actual programming languages. As a result, my research is usually innovative in making hitherto unseen connections between these foundational disciplines and more mainstream computer science areas such as rewriting, functional programming, computational algebra and artificial intelligence. I also regard my role as not simply to prove results, nor just to write papers, but to build a substantial body of evidence to substantiate the ideas which underpin the foundational approach I advocate. The following is a brief summary of my research to date:

- **Category Theory:** Category theory is a relatively new mathematical discipline which 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 my research here.
- **Rewriting:** Rewriting Systems are widely used as abstract models of computation as they are computationally expressive while retaining a relatively simple and concrete syntax. My thesis developed the subject of Eta-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 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.
- **Lambda Calculus:** The lambda-calculus is of foundational importance within computer science and, in particular, can be viewed as a paradigmatic functional programming language. I 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 in a uniform manner. I developed a new approach to monad composition which is general in that nearly all monads compose, mathematically elegant in using the standard categorical tools underpinning monads and computationally expressive in supporting canonical recursion operators.

  My research on containers has provided new and suprisingly 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. We are 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 have recently been awarded an EPSRC grant for 230,000 to further this research.

  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 has 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.

- **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 internation 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 loose this desire if the 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 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. The course I taught last year at Nottingham was rated the second most popular in the school by the students. I have taught the following courses:

C01003	Yr 1, University of Leicester	Programme 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
CS203	Yr 2, University of Strathclyde	Topics in Computer Science

Reseach 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. In total, I have published 13 journal papers, 29 conference papers, edited one conference proceedings and have two journal papers under consideration.

Journal Papers

- How I Learned to Stop Worrying and Love Kan Extensions. Neil Ghani and Patricia Johann. Submitted for publication in *Journal of Functional Programming*, 2007.

- Programming with Nested Types: A Principled Approach. Neil Ghani and Patricia Johann. Submitted for publication in *Journal of Higher Order Smbolic Computation*, 2007.
- A Universe of Strictly Positive Families. Neil Ghani, Peter Morris and Thorsten Altenkirch. Accepted for publication in *International Journal of Foundations of Computer Science*, 2007.
- Monadic Augment and Generalised Short Cut Fusion. Neil Ghani and Patricia Johann. In *Journal of Functional Programming*, vol. 17(6), pages 731 - 776, 2007.
- Explicit Substitutions. Neil Ghani, Tarmo Uustalu and Makoto Hamana. In *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. In *Journal of Symbolic Computation*, vol. 41(5), pages 573-590, 2006.
- δ for Data — Differentiating Data Structures. Neil Ghani, Michael Abbott, Thorsten Altenkirch and Conor McBride. In *Fundamentae Informatica*, vol. 65(1,2), pages 1-28, 2005.
- Containers - Constructing Strictly Positive Types. Neil Ghani, Michael Abbott and Thorsten Altenkirch. In *Journal of Theoretical Computer Science*, vol. 341(1), pages 3-27, 2005.
- Monads of Coalgebras: Rational Terms and Term Graphs. Neil Ghani, Christoph Luth and Federico De Marchi. In *Journal of Mathematical Structures in Computer Science*, vol. 15(3), pages 433-451, 2005.
- Coproducts of Ideal Monads. Neil Ghani and Tarmo Uustalu. In *Journal of Theoretical Informatics and Applications*, vol. 38(4), pages 321-342, 2004.
- Rewriting via Coinserters. Neil Ghani and Christoph Luth. In *Nordic Journal of Computing*, vol. 10(4), pages 290-312, 2003.
- Solving Algebraic Equations using Coalgebra. Neil Ghani, Christoph Luth and Federico De Marchi. In *Journal of Theoretical Informatics and Applications*, vol. 37, pages 301-314, 2003.
- Dualizing Initial Algebras. Neil Ghani, Christoph Luth, Federico De Marchi and John Power. In *Journal of Mathematical Structures in Computer Science*, vol. 13(1), pages 349-370, 2003.
- Linear Explicit Substitutions. Neil Ghani, Valeria de Paiva and Eike Ritter. In *Journal of the International Group on Programming Languages*, vol. 8(1), pages 7-31, 2000.
- The Virtues of Eta-Expansion. Neil Ghani and Barry Jay. In *Journal of Functional Programming*, vol. 5(2), pages 135-154, 1996.

Editing of Conference Proceedings

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

Internationally Reviewed Conference Papers

- Foundations for Structured Programming with GADTs. Neil Ghani and Patricia Johann. Accepted for publication in *Procs. of Principles of Programming Languages 2008*.
- Initial Algebra Semantics is Enough! Neil Ghani and Patricia Johann. In *Procs. of Typed Lambda Calculus and Applications 2007*, Lecture Notes in Computer Science, number 4583, pages 207-222, Springer-Verlag, 2007.
- Constructing Strictly Positive Families. Neil Ghani, Peter Morris and Thorsten Altenkirch. In *Procs. of Australasian Symposium on Theory of Computing 2007*, ACM International Conference Proceeding Series, vol. 240, pages 111-121, Australian Computer Society, 2007.
- Continuous Functions on Final Coalgebras. Neil Ghani, Peter Hancock and Dirk Pattinson. In *Procs. of Coalgebraic Methods in Computer Science 2006*, Electronic Notes in Theoretical Computer Science, vol. 164, pages 141-155, Elsevier, 2006.
- Representing cyclic structures as nested datatypes. Neil Ghani, Makoto Hamana, Tarmo Uustalu and Varmo Vene. In *Procs. of Trends in Functional Programming 2006*, pages 173-188, Intellect, 2006.
- Monadic augment and generalised short cut fusion. Neil Ghani, Patricia Johann, Tarmo Uustalu and Varmo Vene. In *Procs. of International Conference on Functional Programming 2005*, ACM SIGPLAN Notices, vol. 40(9), pages 294-305, ACM Press, 2005.
- Generalising the Augment Combinator. Neil Ghani, Tarmo Uustalu and Varmo Vene. In *Procs. of Trends in Functional Programming 2005*, pages 65-78, Intellect, 2006.
- Abstract Modularity. Neil Ghani, Michael Abbott and Christoph Lüth. In *Procs. of Rewriting Techniques and Applications 2005*, Lecture Notes in Computer Science number 3467, pages 46-60, Springer-Verlag, 2005.
- Relationally Staged Computation in the π-calculus. Neil Ghani, Bjorn Victor and Kidane Yemane. In *Procs. of Coalgebraic Methods in Computer Science 2004*, Electronic Notes in Theoretical Computer Science, vol. 106, pages 105-120, Elsevier, 2004.
- Constructing Programs with Quotient Types. Neil Ghani, Michael Abbott, Thorsten Altenkirch and Conor McBride. In *Procs. of Mathematics of Programme Construction 2004*, Lecture Notes in Computer Science, number 3125, pages 2-15, Springer-Verlag, 2004.
- Representing Nested Inductive Types using W-types. Neil Ghani, Michael Abbott and Thorsten Altenkirch. In *Procs. of International Colloquium on Automata, Languages and Programming 2004*, Lecture Notes in Computer Science, number 3142, pages 59-71, Springer-Verlag, 2004.
- Build, Augment, Destroy. Universally. Neil Ghani, Tarmo Uustalu and Varmo Vene. In *Procs. of Asian Symposium on Programming Languages and Systems 2004*, Lecture Notes in Computer Science, number 3302, pages 327-341, Springer-Verlag, 2004.

- Difunctorial Semantics of Object Calculus. Neil Ghani and Johan Glimming. In *Procs. of Workshop on Object-Oriented Development 2004*, Electronic Notes in Theoretical Computer Science, vol. 138(2), pages 79-94, Elsevier, 2005.

- Categories of Containers. Neil Ghani, Michael Abbott and Thorsten Altenkirch. In *Procs. of Foundations of Software Science and Computation Structures 2003*, Lecture Notes in Computer Science, number 2620, pages 23-38, Springer-Verlag, 2003.

- Derivatives of Containers. Neil Ghani, Michael Abbott, Thorsten Altenkirch and Conor McBride. In *Procs. of Typed Lambda Calculus and Applications 2003*, Lecture Notes in Computer Science, number 2701, pages 16-30, Springer-Verlag, 2003.

- A Rewriting Alternative to Reidemeister Schreier. Neil Ghani and Anne Heyworth. In *Procs. of Rewriting Techniques and Applications 2003*, Lecture Notes in Computer Science, number 2706, pages 452-466, Springer-Verlag, 2003.

- Explicit Substitutions and Higher-Order Syntax. Neil Ghani and Tarmo Uustalu. In *Procs. of Mechanized Reasoning about Languages with Variable Binding 2003*, pages 1-8, ISBN: 1-58113-800-8, ACM Press, 2003.

- Computing over K-modules. Neil Ghani and Anne Heyworth. In *Procs. of Computing: The Australasian Theory Symposium 2002*, Electronic Notes in Theoretical Computer Science, vol. 61, pages 34-50, Elsevier, 2002.

- Monads and Modularity. Neil Ghani and Christoph Lüth. In *Procs. of Frontiers of Combining Systems 2002*, Lecture Notes in Artificial Intelligence, number 2309, pages 18-32, Springer-Verlag, 2002.

- Coalgebraic Monads. Neil Ghani, Christoph Lüth and Federico De Marchi. In *Procs. of Coalgebraic Methods in Computer Science 2002*, Electronic Notes in Theoretical Computer Science, vol. 65(1), pages 71-91, Elsevier, 2002.

- Composing Monads Using Coproducts. Neil Ghani and Christoph Lu̇th. In *Procs. of International Conference on Functional Programming 2002*. ACM SIGPLAN Notices, vol. 37(9), pages 294-305, ACM Press, 2002.

- Algebras, Coalgebras, Monads and Comonads. Neil Ghani, Christoph Lüth, Federico de Marchi and John Power. In *Procs. of Coalgebraic Methods in Computer Science 2001*, Electronic Notes in Theoretical Computer Science, vol. 44(1), pages 128-145, Elsevier, 2001.

- Categorical Models of Explicit Substitutions. Neil Ghani, Valeria de Paiva and Eike Ritter. In *Procs. of Foundations of Software Science and Computation Structures 1999*, Lecture Notes in Computer Science, number 1578, pages 197-211, Springer-Verlag, 1999.

- Explicit Substitutions for Constructive Necessity. Neil Ghani, Valeria de Paiva and Eike Ritter. In *Procs. of International Colloquium on Automata, Languages and Programming 1998*, Lecture Notes in Computer Science, number 1443, pages 743-754, Springer-Verlag, 1998.

- Monads and Modular Term Rewriting. Neil Ghani and Christoph Lüth. In *Procs. of Category Theory and Computer Science 1997*, Lecture Notes in Computer Science, number 1290, pages 69-86, Springer-Verlag, 1997.

- On Modular Properties of Higher Order Extensional Lambda Calculi. Neil Ghani and Roberto Di Cosmo. In *Procs. of International Colloquium on Automata, Languages and Programming 1997*, Lecture Notes in Computer Science, number 1256, pages 237-247, Springer-Verlag, 1997.
- Eta-Expansions in Dependent Type Theory — The Calculus of Constructions. Neil Ghani. In *Procs. of Typed Lambda Calculus and Applications 1997*, Lecture Notes in Computer Science, number 1210, pages 164-180, Springer-Verlag, 1997.
- Eta-Expansions in F^w. Neil Ghani. In *Procs. of Computer Science Logic 1996*, Lecture Notes in Computer Science, number 1258, pages 182-197, Springer-Verlag, 1996.
- Beta-Eta Equality for Coproducts. Neil Ghani. In *Procs. of Typed Lambda Calculus and Applications 1995*, Lecture Notes in Computer Science, number 902, pages 171-185, Springer-Verlag, 1995.

Thesis

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

Research Supervision

Rawle Prince,	PhD, University of Nottingham,	since 2006
Mauro Jaskelioff,	PhD, University of Nottingham,	since 2005
Michael Abbott,	PhD, University of Leicester,	2000-2003
Federico de Marchi,	PhD, University of Leicester,	1999-2002
Peter Hancock,	RA, Universities of Nottingham,	since 2005
Anne Heyworth,	RA, University of Leicester,	2001-2004

Academic Service

I am the head of the *Mathematically Structured Programming* group at the Univsersity of Strathclyde and also on the research committee of our department. At Nottingham I was in charge of the organisation of our 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 School and I was the course director for one of the degrees that we offered. I have demonstrated my ability to discharge senior administrative roles, e.g. 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 the 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, courses etc.

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 serve on the Engineering and Physical Sciences Research Council (EPSRC) Peer Review College. 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 which I was elected to by my fellow academics.

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 was on the programme committee of RTA in 2002 and CMCS in 2006, and will be on the programme committee of RTA in 2008 and CMCS in 2008. II 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