Curriculum Vitae

Gavin Bierman

7 Pretoria Road
Cambridge CB4 1HD
UK

Home: +44 1223 526 712
Mobile: +44 7817 915 431
Email: gavin.bierman@oracle.com
Website: http://gavinbierman.github.io
https://labs.oracle.com/pls/apex/f?p=labs:bio:0:2044

Current Employment

Consulting member of technical staff Java Platform Group and Oracle Labs, Oracle
Mar 2014 – Current

Work History

Senior Researcher Microsoft Research Cambridge
Sept 2008 – Feb 2014
Researcher Microsoft Research Cambridge
March 2004 – Aug 2008
University Lecturer University of Cambridge Computer Laboratory
Oct 2000 – Feb 2004
College Lecturer & Director of Studies St John's College, Cambridge
Oct 2000 – Feb 2004
University Lecturer University of Warwick, Department of Computer Science
Oct 1999 – Sept 2000
Research Fellow Gonville and Caius College, Cambridge
Oct 1995 – Sept 1999
EPSRC Research Fellow University of Cambridge Computer Laboratory
Dec 1993 – Sept 1995
Junior Research Fellow Wolfson College, Cambridge
Oct 1993 – Sept 1995

Education

PhD University of Cambridge Computer Laboratory
Oct 1990 – Dec 1993 Title: On Intuitionistic Linear Logic
BSc(Eng) Hons. First Class. Imperial College, University of London
Oct 1987 – Jun 1990 Computing.

Publications

1. What is a Secure Programming Language? (with Cifuentes). Proceedings of SNAPL 2019.
2. Self-managed collections: Off-heap memory management for scalable query-dominated collections. (with Nagel, Dragojevic, and Viglas). In Proceedings of EDBT 2017.
3. Processing Declarative Queries Through Generating Imperative Code in Managed Runtimes. (with Viglas and Nagel). Proceedings of ICDE 2017.
4. Safe & Efficient Gradual Typing for TypeScript. (with Rastogi, Swamy, Fournet, and Vekris). Proceedings of POPL 2015.
5. Understanding TypeScript (with Abadi and Torgersen). Proceedings of ECOOP 2014.
6. Polymonadic Programming. (with Hicks, Guts, Leijen, and Swamy) Proceedings of MSFP 2014.
7. Gradual typing safely embedded in JavaScript. (with Bhargavan, Bierman, Chen, Fournet, Rastogi, Strub, Swamy) Proceedings of POPL 2014.
8. Separation logic for object-oriented programming. (with Parkinson). Invited chapter in "Aliasing in object-oriented programming". LNCS 7850. Pages 366-406. January 2013. Springer-Verlag.
9. Pause 'n' play: Formalizing asynchronous C#. (with Russo, Mainland, Meijer and Torgersen). Proceedings of ECOOP 2012.
10. Extending the relational algebra with similarities. (with Hajdinjak). Mathematical Structures in Computer Science, 22(4):686–718. 2012. Cambridge University Press
11. Semantic subtyping with an SMT solver. (with Gordon, Hritcu and Langworthy). Journal of Functional Programming, 22(1):31-105. March 2012. Cambridge University Press.
12. A co-relational model of data for large shared data banks. (with Meijer). ACM Queue, Volume 9, Number 3. March 2011.
13. Semantic subtyping with an SMT solver. (with Gordon, Hritcu and Langworthy). Microsoft Research Technical Report MSR-TR-2010-99. December 2010.
14. Semantic subtyping with an SMT solver. (with Gordon, Hritcu and Langworthy). Proceedings of ICFP 2010.
15. Adding dynamic types to C#. (with Meijer and Torgersen). Proceedings of ECOOP 2010.
16. A theory of typed coercions and its applications. (with Hicks and Swamy) Proceedings of ICFP 2009

17. From Java to UpgradeJ: An empirical study. (with Tempero, Parkinson and Noble) Proceedings of Workshop on Hot Topics in Software Updating. 2008
18. UpgradeJ: Incremental typechecking for class upgrades. (with Parkinson and Noble) Proceedings of ECOOP 2008.
19. Dynamic Rebinding for Marshalling and Update, via Redex-time and Destruct-time Reduction. (with Hicks, Sewell, Stoyle and Wansborough) Journal of Functional Programming, Volume 18, issue 04, pages 437-502. July 2008.
20. Selected papers from the 10th International Symposium on Database Programming Languages (DBPL 2005). (co-edited with Koch). Information Systems, Volume 33, Issue 4-5 (Special edition). June 2008.
21. UpgradeJ: Incremental typechecking for class upgrades (long version). (with Parkinson and Noble) University of Cambridge Computer Laboratory Technical Report 716. April 2008. 33+iipp.
22. Separation logic, abstraction and inheritance. (with Parkinson) Proceedings of POPL 2008.
23. Lost in translation: Formalizing proposed extensions to C# (with Meijer and Torgersen) Proceedings of OOPSLA 2007.
24. Mutatis Mutandis: Safe and predictable dynamic software updating (Journal Version). (with Stoyle, Hicks, Sewell and Neamtiu) ACM Transactions on Programming Languages and Systems. Volume 29, issue 4, article 22. August 2007. 70pp.
25. Formalizing and extending C# type inference. Proceedings of FOOL/WOOD 2007.
26. LINQ: Reconciling objects, relations and XML in the .NET framework. (with Meijer and Beckman) Proceedings of SIGMOD 2006.
27. Report on 10th International Symposium on Database Programming Languages (DBPL 2005). (with Koch). SIGMOD Record, 35(1). March 2006. 6pp
28. Proceedings of 10th International Symposium on Database Programming Languages (DBPL 2005). (edited with Koch) Volume 3774 of Springer LNCS.
29. First-class relationships in an object-oriented language (Extended version). (with Wren) University of Cambridge Computer Laboratory Technical Report 642. August 2005. 53pp.
30. The essence of data access in Cω. (with Meijer and Schulte) Proceedings of ECOOP 2005.
31. First-class relationships in an object-oriented language. (with Wren) Proceedings of ECOOP 2005.
32. Unifying tables, objects and documents. (with Meijer and Schulte). Proceedings of DP-COOL 2005. Volume 27 of John von Neumann Institute of Computing.
33. First-class relationships in an object-oriented language. (with Wren) Proceedings of FOOL 2005.
34. Mutatis Mutandis: Safe and predictable dynamic software updating. (with Stoyle, Hicks, Sewell and Neamtiu) Proceedings of POPL 2005.
35. Separation logic and abstraction. (with Parkinson) Proceedings of POPL 2005.
36. Dynamic rebinding for marshalling and update, with destruct-time lambda. (with Hicks, Sewell, Stoyle and Wansborough) University of Cambridge Computer Laboratory Technical Report 568. February 2004. 83+ii pp.
37. Programming with circles, triangles and rectangles. (with Meijer and Schulte) Proceedings of XML 2003.
38. Ubiquitous Data. (with Buneman and Gardner) Position paper presented at UK-UbiNet Workshop 2003. September 2003.
39. Dynamic rebinding for marshalling and update, with destruct-time lambda. (with Hicks, Sewell, Stoyle and Wansborough) Proceedings of ICFP 2003.
40. Formal semantics and analysis of object queries (Extended Abstract). Proceedings of SIGMOD 2003.
41. MJ: An imperative core calculus for Java and Java with effects. (with Parkinson and Pitts) University of Cambridge Computer Laboratory Technical Report 563. April 2003. 51+ii pp.
42. Effects and effect inference for a core Java calculus. (with Parkinson) Proceedings of WOOD 2003. Appears in ENTCS 82(8).
43. Formalizing dynamic software updating (Extended Abstract). (with Hicks, Sewell and Stoyle) Proceedings of Workshop on Unexpected Software Evolution (USE 2003).
44. Iota: A concurrent, XML scripting language with applications to Home-Area Networks. (with Sewell) University of Cambridge Computer Laboratory Technical Report 557. January 2003.
45. Inferring the principal type and schema requirements of an OQL query. (with Trigoni) Proceedings of BNCOD 2001.
46. Strong normalisation of cut-elimination in classical logic. (with Urban) Fundamenta Informaticae. 45(1-2):123-155. January 2001.
47. On an Intuitionistic Modal Logic. (with de Paiva) Studia Logica. 65(3):383-416. 2000.
48. Operational properties of Lily, a polymorphic linear lambda calculus with recursion. (with Pitts and Russo) Proceedings of HOOTS 2000. Volume 41 of Electronic Notes in Theoretical Computer Science, Elsevier. September 2000.
49. Towards a formal type system for ODMG OQL. (with Trigoni) University of Cambridge Computer Laboratory Technical Report 497. September 2000. 20pp.
50. Using XML as an object interchange format. Proposal published on the ODMG website.
51. Program equivalence in a linear functional language. Journal of Functional Programming. 10(2):167-190. 2000.
52. A classical linear lambda calculus. Theoretical Computer Science. 227(1-2):43-78. 1999.
53. Strong normalisation of cut elimination in classical logic. (with Urban) Proceedings of TLCA 1999.
54. Multiple modalities. University of Cambridge Computer Laboratory Technical Report 455. December 1998. 26+ii pp.
55. A computational interpretation of the λμ-calculus. University of Cambridge Computer Laboratory Technical Report 448. September 1998. 27+ii pp.
56. A computational interpretation of the λμ-calculus. Proceedings of MFCS 1998.
57. Linear logic. In Routledge Encyclopedia of Philosophy. 2200 words. Published July 1998.
58. Computational types from a logical perspective. (with Benton and de Paiva). Journal of Functional Programming, 8(2):177-193. March 1998
59. A new general purpose parallel database system. (with Afshar, Bates and Moody). Proceedings of IEEE International Symposium on Parallel Architectures, Algorithms and Networks. Pages 2-8. December 1997.
60. Observations on a linear PCF. University of Cambridge Computer Laboratory Technical Report 412. January 1997. 30pp.
61. Towards a classical linear lambda-calculus. Proceedings of Tokyo Meeting on Linear Logic. Volume 3 of Electronic Notes in Theoretical Computer Science , Elsevier. Eds J.-Y. Girard, M. Okada and A. Scedrov. November 1996. 13pp.
62. A note on full intuitionistic linear logic. Annals of Pure and Applied Logic, 79(3):281-287. 1996.
63. A classical linear lambda-calculus. University of Cambridge Computer Laboratory Technical Report 401. July 1996. 41pp.

64. Intuitionistic necessity revisited. (with de Paiva). Technical Report CSRP-96-10, School of Computer Science, University of Birmingham. June 1996. 18pp.
65. Computational types from a logical perspective I. (with Benton and de Paiva). University of Cambridge Computer Laboratory Technical Report 365. May 1995.
66. What is a categorical model of intuitionistic linear logic? Proceedings of TLCA 1994.
67. On intuitionistic linear logic. PhD Thesis, University of Cambridge Computer Laboratory, December 1993. Available as Technical Report 346, August 1994.
68. A term calculus for intuitionistic linear logic. (with Benton, de Paiva and Hyland). Proceedings of TLCA 1993.
69. Linear lambda-calculus and categorical models revisited. (with Benton, de Paiva and Hyland). Proceedings of CSL 1993.
70. Intuitionistic necessity revisited (extended abstract). (with de Paiva). In Proceedings of Applied Logic Conference, December 1992.
71. Term assignment for intuitionistic linear logic. (with Benton, de Paiva and Hyland). Technical Report 262, University of Cambridge Computer Laboratory. August 1992.

PATENTS GRANTED

1. Accessing a migrated member in an updated type (with Goetz and Rose) US Patent US10,908,886. Granted April 28, 2020.
2. Overriding a migrated method in an updated type (with Goetz and Rose). US Patent US10,635,420. Granted April 28, 2020.
3. Flow-based scoping. (with Goetz and Steele). US Patent No. 10,310,827. Granted June 4, 2019.
4. Semantic subtyping for declarative data scripting language by calling a theorem prover. (with Langworthy, Gordon, Box, Lovering, Schlimmer, Doty). US Patent No. 8,413,119. Granted April 2, 2013.
5. Compositional lifting of operations over structural types. (with Meijer and Schulte). US Patent No. 7,912,863. Granted March 22, 2011.
6. Type inference for object-oriented languages. (with Meijer, Torgersen, Hejlsberg, Van Velzen, Hallam, Lippert, Warren, Vick, and Silver). US Patent No. 7,873,592. Granted January 18, 2011.
7. Type-based extensions for object-oriented languages based on coercive subtyping with restrictions. (with Meijer and Schulte). US Patent no. 7,774,376. Granted August 10, 2010.

PHD STUDENTS SUPERVISED

Alisdair Wren University of Cambridge, 2007
Relationships for OO languages

Gareth Stoyle University of Cambridge, 2006
A Theory of Dynamic Software Updates

Matthew Parkinson University of Cambridge, 2005
Local Reasoning for Java

Christian Urban University of Cambridge, 2000
Classical Logic and Computation

PHD EXAMINATIONS

Luis Gabriel Ganchinho de Pina University of Lisbon 2015
Practical Dynamic Software Updating

Neville Grech University of Southampton, 2013
Preemptive type checking in dynamically typed languages

Melita Hajdinjak University of Ljubljana, 2012
Relational algebra with similarities

Conglun Yao University of Birmingham, 2010
Strongly-typed, compile-time safe and loosely coupled data persistence

Susanne Cech Previtali ETH, Zurich, 2009
Dynamic updating of object-oriented software systems based on aspects

Rok Strniša University of Cambridge, 2009
Formalising, improving and reusing the Java module system

Alex Summers Imperial College, 2008
Curry-Howard term calculi for Gentzen-style classical logics

Viktor Vafeiadis University of Cambridge, 2007
Modular fine-grained concurrency verification

Francisco Alberti University of Paris, 2005
Analyse Statique Typé des Propriétés Structurelles des Programmes

András Belokosztolski University of Cambridge, 2004
Role-based access control policy administration

Lucian Wischik University of Cambridge, 2002
Explicit Fusions: theory and implementation

Jacob Howe University of St Andrews, 1998
Proof search issues in some non-classical logics

COMMUNITY SERVICE

OOPSLA 2021 [PC member]; OOPSLA 2019 [PC member]; POPL 2019 [PC member]; ESOP 2015 [PC member]; ESOP 2014 [PC member]; SPLASH 2013 workshops [PC member]; ECOOP 2013 [PC member]; ECOOP 2012 [PC member]; DBPL 2011 [PC member]; FTfJP 2011 [PC member]; ECOOP 2011 [PC member]; STOP 2011 [PC member]; MPOOL 2010 [PC member]; RAOOL 2009 [PC member]; DBPL 2009 [PC member]; HotSWUp 2009 [PC member]; PLAN-X 2009 [co-chair]; ECOOP 2008 [PC member]; HotSWUp 2008 [PC member]; IMLA 2008 [PC member]; RAOOL 2008 [PC member]; PLAN-X 2006 [PC member]; FOOL/WOOD 2006 [PC member]; DBPL 2005 [co-chair]; MPOOL 2005 [PC member]; APPSEM 2004 [PC member]