

March 6, 2026

Brief Bio. Peter O’Hearn is a researcher at Meta Superintelligence Labs and a Professor at University College London. He has made significant contributions to programming languages, logic, and software verification. Peter developed Separation Logic and Incorrectness Logic, theories which have been used in various reasoning tools, including Infer, a program analyzer that has detected hundreds of thousands of bugs at Facebook and has been used in other companies including Microsoft and Amazon. Prior to joining Meta, Peter co-founded Monoidics, a verification startup that developed Infer and was acquired by Facebook in 2013. Peter has received numerous awards for his work, including the Gödel Prize and being elected Fellow of the Royal Society.

Awards, Honours, etc

- **Gödel Prize, 2016.** For the invention of Concurrent Separation Logic. (with S Brookes)
- **Fellow of the Royal Society**, elected 2018
- **Fellow of the Royal Academy of Engineering**, elected 2016
- **Fellow of the ACM**, elected 2025
- **IEEE Cybersecurity Award for Practice, 2021.** For scaling advanced program analysis to industrial practice. (With D Distefano, M Fahndrich, F Logozzo)
- **CAV Award, 2016.** For the development of Separation Logic and for demonstrating its applicability in the automatic verification of programs that mutate data structures. (With J Reynolds, J Berdine, S Ishtiaq, C Calcagno, D Distefano, H Yang)
- **CONCUR Test of Time Award, 2024.** For my CONCUR’04 paper “Resources, Concurrency and Local Reasoning” along with “A Semantics for Concurrent Separation Logic” by S Brookes.
- **Most Influential POPL Paper (test-of-time) Award, 2019.** For POPL’09 paper “Compositional Shape Analysis by means of Bi-Abduction”. (With C Calcagno, D Distefano, H Yang)
- **Most Influential POPL Paper (test-of-time) Award, 2011.** For POPL’01 paper “BI as an Assertion Language for Mutable Data Structures” (With S Ishtiaq)
- **Distinguished Paper Award, OOPSLA 2022.** For “Finding Real Bugs in Big Programs with Incorrectness Logic”. (With Q Lee and others)
- **Doctor of Laws Honoris Causa (Honorary Doctorate), Dalhousie University, 2018**
- **O’HearnFest:** <https://popl24.sigplan.org/home/ohearn-fest>.

Industry Positions

- Meta Superintelligence Labs, 2024-present. AI Researcher.
- Lacework UK, 2021-2024. Researcher+Engineer.
- Facebook UK Ltd., 2013-2021. Researcher+Engineer.
- Monoidics Ltd. Co-founder, 2009-2013. Startup based on academic static analysis work (SpaceInvader, Abductor). Monoidics was acquired by Facebook in 2013.

Academic Positions

- University College London. Professor of Computer Science, 2012-present. (Part time since 2013.)
- Queen Mary, University of London. Professor 1999-2012; Reader 1996-1999
- Syracuse University. Assistant Professor, 1990-1995

Main Contributions

- **Separation Logic** (with JC Reynolds and others, [11,63,64]), attacking 30-year open problem of tractable reasoning about data structures in computer memory. I discovered a theory of *local reasoning*, leading to scalability of automated reasoning to large codebases. This has had impact via the Infer static analyzer, in use at FB, Microsoft, Amazon and other companies [FB1-4, Amzn1,Msft1,Sona].

I also developed **Concurrent Separation Logic** [38,Godel], which advanced a new approach to modular reasoning about shared-memory concurrent programs. CSL is has had impact in systems verification [6,Bed,muK,Amzn3,Msft2], and in providing a foundation for the Rust programming language [Rustb]

- **Separation Logic-based automated analysis tools**, achieving some of the leading results on automatically proving properties of mutable data structures in real-world systems code in tools such as SPACE INVADER and ABDUCTOR [34,36,47,48,52,Cav]
- Co-founded **Monoidics Ltd** in 2009, which marketed a software verification tool, INFER, based on Separation Logic. Monoidics was acquired by Facebook in 2013 [Monoidics], and as part of the deal I took a position at Facebook.
- **Bunched Logic** (with DJ Pym, [67]), a novel logic of resources. Separation Logic builds on Bunched Logic.
- **Relational Semantics of Local State** (with RD Tennent [74]), combining functor categories and relations to describe mathematical principles underlying hidden state in programs
- **Facebook Infer static analyzer, fbinfer.com** Led/supported the team that developed FB Infer, an automatic reasoning tool that in the four years to March 2018 had seen over 100,000 of its reported issues fixed by Facebook developers before reaching production, affecting the the main Facebook app, Instagram, Messenger, and WhatsApp [10]. Infer was made open source in 2015 [C,G], and is used by other companies including Amazon [Amzn1,Amzn2], Microsoft [Msft1], Spotify, Uber, JD.com, Sky and Marks and Spencer.
- **RacerD** After leading the Infer team for four years, I moved to a research/engineering role to concentrate on technical work, beginning with static concurrency analysis (a longstanding open problem). The resulting RacerD concurrency analysis [12,13,14,FB2] has some of the most reported impact from a static analysis for concurrency to date: It saw 2,500 fixes of data race issues in the year to March 2018, and was instrumental in the conversion of Facebook’s Android app from a single-threaded to a multi-threaded architecture which resulted in performance and engagement improvements: “Without Infer, Multi-threading in NewsFeed would not have been tenable [11].”
- **Incorrectness Logic** ([9]). I created a new theory, Incorrectness Logic, a kind of dual to Hoare’s logic of correctness aimed at underpinning tools for bug catching [6]. Incorrectness logic is in production at FB [5] and at Microsoft [Msft1].
- **Lacework Code Security** ([LW1,2]). Led the creation of a suite of Code Security products for Lacework, encompassing several varieties of static and dynamic program analysis.

Selected Impact References

- LW1 “Lacework Extends Security Reach Into Application Development.” Michael Vizard. *Security Boulevard*, 14 November 2023.
- LW2 “Lacework Expands into Full Application Lifecycle Protection.” Steve McDowell. *Forbes*, 14 November 2023.
- LW3 “SAST for the people.” Lacework blog post, July 2024. <https://www.lacework.com/blog/sast-for-the-people>

- FB1 “Facebook’s Code Checker.” **Computerphile, Oct 2020**. Non-specialist interview tracing steps from Turing and undecidability to Separation Logic and to Facebook Infer. <https://www.youtube.com/watch?v=tKR2UZdRpV0>. >100k views.
- FB2 “Facebook Engineering Takes a Bite out of Concurrency with RacerD.” Michelle Gienow, **TheNewStack, 19 October 2017**
- FB3 “Facebook’s AI Tool for Squashing Bugs is now Open to All.” Klint Finley, **Wired, 11 June 2015**
- FB4 *fbinfer.com*. WEB PAGE SHOWING INFER’S OPEN-SOURCE USERS INCLUDING AMAZON, MICROSOFT, SPOTIFY, UBER, MOZILLA, JD.COM AND OTHERS.
- Monoidics “Facebook Acquires Assets Of UK Mobile Bug-Checking Software Developer Monoidics.” Josh Constine, **Techcrunch, 18 July 2013**
- Godel “2016 Godel Prize Recognizes Major Advances in Verification of Concurrent Programs.” <https://www.acm.org/media-center/2016/may/goedel-prize-2016>
- Cav “Four Facebook Employees Win the Prestigious CAV Award.” Bryan O’Sullivan. **research.fb.com blog, 5 Sept 2016** (STATING OVER 1000 BUGS/MONTH FIXED)
- Msft1 “Slaying Zombie ‘No Repro’ Crashes with Infer#.” Xin Shi. <https://devblogs.microsoft.com/dotnet/slaying-zombie-no-repo-crashes-with-infersharp//>
- Msft2 “SteelCore: an extensible concurrent separation logic for effectful dependently typed programs.” N Swamy, A Rastogi, A Fromherz, D Merigoux, D Ahman, G Martínez: Proc. ACM Program. Lang. 4(ICFP): 121:1-121:30 (2020)
- Amzn1 “How automated reasoning improves the Prime Video experience.” Franco Raimondi, Bor-Yuh Evan Chang. <https://www.amazon.science/blog/how-automated-reasoning-improves-the-prime-video-experience>. April 19, 2021.
- Amzn2 “Amazon CodeGuru now includes recommendations powered by Infer.” <https://aws.amazon.com/about-aws/whats-new/2021/10/amazon-codeguru-recommendations-infer/>. Oct 4, 2021.
- Amzn3 “Formally Verifying FreeRTOS’ Interprocess Communication Mechanism.” Nathan Chong and Bart Jacobs. 2021. Embedded World Exhibition and Conference, 2021.
- Sona “Sonatype Lift Integrates Facebook Infer, Google ErrorProne, and Other Code Analyzers.” June 21, 2021. InfoQ. <https://www.infoq.com/news/2021/06/Sonatype-lift-code-analysis/>
- Bed “Towards an Axiomatic Basis for C++.” G Malecha, A Anand, and G Stewart. Coq2020. https://coq-workshop.gitlab.io/2020/abstracts/Coq2020_04-04-axiomatic-cpp.pdf
- muK F Xu, M Fu, X Feng, X Zhang, H Zhang, Z Li: A Practical Verification Framework for Preemptive OS Kernels. CAV (2) 2016: 59-79
- Rustb R Jung, J-H Jourdan, R Krebbers, D Dreyer: RustBelt: securing the foundations of the Rust programming language. Proc. ACM Program. Lang. 2(POPL): 66:1-66:34 (2018)

Research Publications

1. Logic.py: Bridging the Gap between LLMs and Constraint Solvers. Pascal Kesseli, Peter W. O’Hearn, Ricardo Silveira Cabral: *Neurips 2025*
2. Harden and Catch for Just-in-Time Assured LLM-Based Software Testing: Open Research Challenges. Mark Harman, Peter W. O’Hearn, Shubho Sengupta: *FSE 2025*

3. Non-termination Proving at Scale. Azalea Raad, Julien Vanegue, Peter W. O’Hearn: *OOPSLA 2024*
4. A General Approach to Under-Approximate Reasoning About Concurrent Programs. Azalea Raad, Julien Vanegue, Josh Berdine, Peter W. O’Hearn: *CONCUR 2023: 25:1-25:17*
5. Finding real bugs in big programs with incorrectness logic. Quang Loc Le, Azalea Raad, Jules Villard, Josh Berdine, Derek Dreyer, Peter W. O’Hearn: *OOPSLA 2022*
6. Concurrent incorrectness separation logic. Azalea Raad, Josh Berdine, Derek Dreyer, Peter W. O’Hearn. *POPL 2022*
7. Applying formal verification to microkernel IPC at Meta. Q Carbonneaux, N Zilberstein, C Klee, P O’Hearn and F Zappa Nardelli. *CPP’22*.
8. Algebra for program correctness and incorrectness. Bernhard Moller, Peter O’Hearn and Tony Hoare. *RAMICS 2021*
9. Local reasoning about the presence of bugs.: Incorrectness Separation Logic. Azalea Raad, Josh Berdine, Hoang-Hai Dang, Derek Dreyer, Peter W. O’Hearn, Jules Villard. *CAV 2020*
10. Incorrectness Logic. PW O’Hearn. *POPL 2020*
11. Scaling Static Analyses at Facebook. D. Distefano, M Fahndrich, F Logozzo and P O’Hearn: *Communications of the ACM, August 2019*
12. Separation Logic. P O’Hearn: *Communications of the ACM, February 2019, vol. 62, no. 2*
13. A True Positives Theorem for a Static Race Detector. N Gorrogiannis, P O’Hearn, I Sergey: *POPL 2019*
14. RacerD: Compositional Static Race Detection. S Blackshear, N Gorrogiannis, P O’Hearn, I Sergey: *OOPSLA 2018*
15. Experience Developing and Deploying Concurrency Analysis at Facebook. P O’Hearn. *SAS 2018*
16. Continuous Reasoning: Scaling the Impact of Formal Methods. P O’Hearn : *LICS 2018*
17. Concurrent separation logic. Stephen Brookes, Peter W. O’Hearn: *SIGLOG News 3(3): 47-65 (2016)*
18. “Moving Fast with Software Verification”. Cristiano Calcagno, Dino Distefano, Jérémy Dubreil, Dominik Gabi, Pieter Hooimeijer, Martino Luca, Peter W. O’Hearn, Irene Papanonstantinou, Jim Purbrick, Dulma Rodriguez: *NASA Formal Methods Symposium 2015: 3-11*
19. “From Categorical Logic to Facebook Engineering”. Peter W. O’Hearn. *LICS 2015: 17-20*
20. “On the relation between Concurrent Separation Logic and Concurrent Kleene Algebra”. Peter W. O’Hearn, Rasmus Lerchedahl Petersen, Jules Villard, Akbar Hussain: *J. Log. Algebr. Meth. Program. 84(3): 285-302 (2015)*
21. “Developments in Concurrent Kleene Algebra”. Tony Hoare, Stephan van Staden, Bernhard Möller, Georg Struth, Jules Villard, Huibiao Zhu, Peter W. O’Hearn. *RAMICS 2014: 1-18*
22. “The Essence of Reynolds”. S Brookes, PW O’Hearn and US Reddy. *Conference Record of the 41st ACM Symposium on Principles of Programming Languages*, pp251-256, San Diego, January 2014.
23. “Proving Nontermination via Safety”. HY Chen, B Cook, C, Fuhs, K, Nimkar, PW O’Hearn. *TACAS 2014, pp156-171*.
24. “A Primer on Separation Logic (and Automatic Program Verification and Analysis)”. PW O’Hearn. In *Software Safety and Security; Tools for Analysis and Verification*. NATO Science for Peace and Security Series, vol 33, pp286-318, June 2012.
25. Verification Condition Generation and Variable Conditions in Smallfoot. J Berdine, C Calcagno, PW O’Hearn. *CoRR abs/1204.4804*: April, 2012

26. Compositional shape analysis by means of bi-abduction. C Calcagno, D Distefano, P O'Hearn, and H Yang. *Journal of the ACM*, 73 pages, December 2011. (Preliminary version appeared in POPL'09.)
27. On Locality and the Exchange Law for Concurrent Processes. CAR Hoare, A Hussain, B Möller, P O'Hearn, R Petersen, G Struth. *CONCUR 2011 - Concurrency Theory - 22nd International Conference*, Springer LNCS 6901, pp 250-264
28. The complexity of abduction for separated heap abstractions. N Gorogiannis, M Kanovich and P O'Hearn. *Proceedings of the 18th Static Analysis Symposium*, September 2011
29. Abstraction for Concurrent Objects. I Filipovic, PW O'Hearn, N Rinetzky and H Yang. *Theoretical Computer Science*, vol 411(51-52), pp4379–4398, December 2010. (Preliminary version appeared in ESOP'09)
30. Blaming the client: on data refinement in the presence of pointers. I Filipovic, P O'Hearn, N Torp-Smith and H Yang. *Formal Aspects of Computing*, vol 22(5), pp 547-583, 2010
31. Verifying linearizability with hindsight. PW O'Hearn, N Rinetzky, MT Vechev, E Yahav, G Yorsh. *Proceedings of the 29th ACM Symposium on Principles of Distributed Computing*, pp85-94, 2010
32. Graphical models of separation logic. I Wehrman, CAR Hoare and PW O'Hearn. *Inf. Process. Lett.* 109(17), pp1001-1004, 2009
33. Separation and Information Hiding. PW O'Hearn, H Yang and JC Reynolds. *ACM Transactions on Programming Languages and Systems* 31(3), 49 pages. April 2009. (Preliminary version appeared in POPL'04)
34. Abstraction for Concurrent Objects. I Filipovic, PW O'Hearn, N Rinetzky and H Yang. *Proceedings of the 18th European Symposium on Programming*, Springer Lecture Notes in Computer Science vol 5502, pp252-266, 2009. (Expanded version appeared in TCS'10)
35. Compositional shape analysis by means of bi-abduction. C Calcagno, D Distefano, P O'Hearn, and H Yang. *Conference Record of the 36th ACM Symposium on Principles of Programming Languages*, pp289-300, Savannah, January 2009. (Expanded version in JACM'11.)
36. Separation Logic Tutorial. PW O'Hearn. *Proceedings of the 24th International Conference on Logic Programming*, Springer Lecture Notes in Computer Science vol 5366, pp15-21, 2008.
37. Scalable Shape Analysis for Systems Code. H Yang, O Lee, J Berdine, C Calcagno, B Cook, D Distefano and P O'Hearn. *20th International Conference on Computer Aided Verification*, Princeton, Springer Lecture Notes in Computer Science 5123, pp385-398, 2008.
38. Separation Logic Semantics for Communicating Processes. T Hoare and P O'Hearn. *Proceedings of 1st International Conference on Foundations of Informatics, Computing and Software*, pp3-25. Shanghai. Electr. Notes Theor. Comput. Sci. 212. 2008
39. Resources, Concurrency and Local Reasoning. PW O'Hearn. *Theoretical Computer Science* 375(1-3), pp271-307. May 2007. (Preliminary version appeared in CONCUR'04, LNCS 3170, 49–67.)
40. Local Action and Abstract Separation Logic. C Calcagno, PW O'Hearn and H Yang. *Proceedings of the 22nd IEEE Symposium on Logic in Computer Science*, pp 366-378, Wroclaw, 2007.
41. Variance Analyses from Invariance Analyses. J Berdine, B Cook, A Chawdhary, D Distefano and P O'Hearn. *Conference Record of the 34nd ACM Symposium on Principles of Programming Languages*, pp211-224. Venice, January 2007.
42. Modular Proof of a Non-blocking Stack. M Parkinson, R Bornat and P O'Hearn. *Conference Record of the 34nd ACM Symposium on Principles of Programming Languages*, pp297-302. Venice, January 2007.
43. Footprint analysis: A shape analysis that discovers preconditions. C Calcagno, D Distefano, P O'Hearn, and H Yang. *Proceedings of the 14th International Static Analysis Symposium*,

44. Shape Analysis for Composite Data Structures. J Berdine, C Calcagno, B Cook, D Distefano, P O’Hearn, T Wies and H Yang. *19th International Conference on Computer Aided Verification*, Berlin, Springer Lecture Notes in Computer Science 4590, pp178–192, 2007.
45. Verified Software: A Grand Challenge. CB Jones, PW O’Hearn and JCP Woodcock. *IEEE Computer* 39(4), pp93-95, 2006.
46. Automatic Termination Proofs for Programs with Shape-shifting Heaps. J Berdine, B Cook, D Distefano and P O’Hearn. In *18th International Conference on Computer Aided Verification*, Seattle, Lecture Notes in Computer Science 4144, pp386–400, 2006.
47. Beyond Reachability: Shape Abstraction in the Presence of Pointer Arithmetic. C. Calcagno, D Distefano, PW O’Hearn and H Yang. In *13th International Static Analysis Symposium*, Seoul Lecture Notes in Computer Science 4134, pp182-203, 2006
48. Smallfoot: Modular Automatic Assertion Checking with Separation Logic. J Berdine, C Calcagno and PW O’Hearn. In *4th Symposium on Formal Methods for Components and Objects*, Amsterdam, Lecture Notes in Computer Science 4111, pp115-137, 2006.
49. A local shape analysis based on separation logic. D Distefano, P O’Hearn and H Yang. *12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, Vienna, Lecture Notes in Computer Science 3920, pp287-302, 2006
50. Strong Update, Disposal, and Encapsulation in Bunched Typing. J Berdine and P O’Hearn. In *Proceedings of the 22nd Conference on Mathematical Foundations of Program Semantics*. pp81-98, 2006.
51. Scalable Specification and Reasoning: Challenges for Program Logic. P O’Hearn. *Verified Software: Theories, Tools, Experiments*, Springer Lecture Notes in Computer Science, vol 4171, pp116-133. Zurich, October 2005.
52. Permission Accounting in Separation Logic. R Bornat, C Calcagno, P O’Hearn and M Parkinson. *Conference Record of the 32nd ACM Symposium on Principles of Programming Languages*, pp259-270. Long Beach, January 2005.
53. Symbolic Execution with Separation Logic. J Berdine, C Calcagno and P O’Hearn. *3rd Asian Symposium on Programming Languages and Systems*. pp52-68, Kyoto, 2005.
54. Separation and Information Hiding. PW O’Hearn, H Yang and JC Reynolds. *Conference Record of the 31st ACM Symposium on Principles of Programming Languages*, pp268-280. Venice, January 2004. (Expanded version appeared in TOPLAS’09)
55. A Decidable Fragment of Separation Logic. J Berdine, C Calcagno and P O’Hearn. *Proceedings of the 24th International Conference on Foundations of Software Technology and Theoretical Computer Science*, Springer Lecture Notes in Computer Science 3328, pp97–109, Chennai, 2004
56. Refinement and Separation Contexts. I Mijajlovic, N Torp-Smith and P O’Hearn, *Proceedings of the 24th International Conference on Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Computer Science 3328, pp421-433, Chennai, 2004
57. Possible worlds and resources: The semantics of BI. DJ Pym, PW O’Hearn, H. Yang. *Theoretical Computer Science* 315/1, pp257-305, 2004.
58. Resources, Concurrency and Local Reasoning. PW O’Hearn. *Proceedings of 15th Conference on Concurrency Theory*, Lecture Notes in Computer Science 3170, pp49-67, 2004. (Prelim version of FCS’07 paper with same title.)
59. Program Logic and Equivalence in the Presence of Garbage Collection. C Calcagno, P O’Hearn, R Bornat. *Theoretical Computer Science*, 298/3, pp557-581, 2003.
60. PW O’Hearn. On Bunched Typing. *Journal of Functional Programming*, 13(4), pp747-796, 2003.
61. J. Berdine, P. O’Hearn, U. Reddy and H. Thielecke. Linear Continuation-Passing. *Higher-Order and Symbolic Computation*, 15(2/3):181–208, September 2002.

62. A Semantic Basis for Local Reasoning. Hongseok Yang and Peter O'Hearn. *5th Conference on Foundations of Software Science and Computer Systems*, Grenoble, Lecture Notes in Computer Science 2303, pp402-416, 2002.
63. Computability and Complexity Results for a Spatial Assertion Language for Data Structures. C Calcagno, H Yang and P O'Hearn. *Proceedings of the Second Asian Workshop on Programming Languages and Systems*, Dajeon, South Korea, pp108-119, 2001.
64. Local Reasoning about Programs that Alter Data Structures. P O'Hearn, J Reynolds, and H Yang. *Proceedings of 15th Annual Conference of the European Association for Computer Science Logic*, Paris, Springer Lecture Notes in Computer Science 2142, pp1-19, 2001.
65. S Ishtiaq and PW O'Hearn. BI as an assertion language for mutable data structures. *Conference Record of the 28th ACM Symposium on Principles of Programming Languages*, pp14-26. London, January 2001.
66. C. Calcagno, PW O'Hearn and S. Ishtiaq. Semantic Analysis of Pointer Aliasing, Allocation and Disposal in Hoare Logic. *Proceedings of the 2nd ACM-SIGPLAN Symposium on Principles and Practice of Declarative Programming*, Montreal, September 2000.
67. On Garbage and Program Logic. C Calcagno and PW O'Hearn. *Proceedings of Fourth Conference on Foundations of Software Science and Computation Structures*, Springer Lecture Notes in Computer Science 2030, pp137-151, 2001.
68. From Algol to polymorphic linear lambda-calculus. PW O'Hearn and JC Reynolds. *Journal of the Association for Computing Machinery*, 47(1), pp167-223, January 2000.
69. Syntactic Control of Interference Revisited. PW O'Hearn, A J Power, M Takeyama and RD Tennent. *Theoretical Computer Science* 228(1-2), October 1999, pp211-252
70. Bireflectivity. PJ Freyd, PW O'Hearn, M Takeyama, AJ Power, R Street and RD Tennent. *Theoretical Computer Science* 228(1-2), October 1999, pp49-76
71. Objects, Interference, and the Yoneda Embedding. PW O'Hearn and US Reddy. *Theoretical Computer Science* 228(1-2), Oct 1999, pp253-282
72. The logic of bunched implications. PW O'Hearn and DJ Pym. *Bulletin of Symbolic Logic*, 5(2), June 1999, pp215-244
73. Resource interpretations, bunched implications and the $\alpha\lambda$ -calculus. PW O'Hearn. *Typed Lambda-Calculi and Applications*, Lecture Notes in Computer Science 1581, pp258-278, L'Aquila, 1999.
74. Polymorphism, Objects and Abstract Types. PW O'Hearn. *SIGACT News*, 29(4), pp39-50, December 1998
75. An axiomatic approach to binary logical relations, with applications to data refinement. Y. Kinoshita, PW O'Hearn, A.J. Power, M. Takeyama, and RD Tennent. *Theoretical Aspects of Computer Software*, Sendai Japan, 1997.
76. Domains and Denotational Semantics: History, Accomplishments and Open Problems. M. Fiore, A. Jung, E. Moggi, P. O'Hearn, J. Riecke, G. Rosolini, I. Stark. *Bulletin of the EATCS*, Vol 59, 1996.
77. Note on Algol and Conservatively Extending Functional Programming. PW O'Hearn. *Journal of Functional Programming*, 6(1), pp171-180, 1996.
78. Kripke logical relations and PCF. PW O'Hearn and JG Riecke. *Information and Computation*, 120(1):107-116, July 1995
79. Parametricity and Local Variables. PW O'Hearn and RD Tennent. *Journal of the Association for Computing Machinery*, 42(3), 658-709, May 1995.
80. A Model for Syntactic Control of Interference. PW O'Hearn. *Mathematical Structures in Computer Science*, vol. 3, pp435-465. 1993.
81. Fully Abstract Translations and Parametric Polymorphism. PW O'Hearn and JG Riecke. *Proceedings of the Fifth European Symposium on Programming*, Springer Lecture Notes in Computer Science 788, pp454-468, 1994

82. Relational Parametricity and Local Variables (preliminary report). PW O’Hearn and RD Tennent. *Conference Record of the 20th ACM Symposium on Principles of Programming Languages*, pp171-184. Charleston, January 1993.
83. Semantical analysis of specification logic, part 2. PW O’Hearn and RD Tennent. *Information and Computation* 107(1), pp25-57, 1993.
84. A resolution framework for finitely-valued first-order logics. P O’Hearn and Z Stachniak. *Journal of Symbolic Computation*. Volume 13, Number 3, pp235-254, March 1992.
85. Semantics of local variables. PW O’Hearn and RD Tennent. *Applications of Categories in Computer Science*, London Math. Soc. lecture notes series 177, pp. 217–238. 1992
86. Linear logic and interference control (preliminary report). PW O’Hearn. in *Category Theory and Computer Science*, Springer Lecture Notes in Computer Science 530, 1991.
87. Resolution in the domain of strongly finite logics. Z Stachniak and P O’Hearn. *Fundamenta Informaticae XIII*, 333–351, 1990.
88. Note on theorem proving strategies for resolution counterparts of non-classical logics. Peter W. O’Hearn and Zbigniew Stachniak. *Proceedings of the ACM-SIGSAM International Symposium on Symbolic and Algebraic Computation*, Portland, 1989

Books Edited

1. G. Leavens, P.W. O’Hearn and S.K. Rajamani, editors. *Verified Software: Theories, Tools, Experiments. Third International Conference*. Springer Lecture Notes in Computer Science, vol 6217, 2010.
2. P.W. O’Hearn and R.D. Tennent, editors. *Algol-like Languages*, volume 1. Birkhauser, Boston, 1997.
3. P.W. O’Hearn and R.D. Tennent, editors. *Algol-like Languages*, volume 2. Birkhauser, Boston, 1998.

Keynotes, Invited Tutorials, etc.

1. “My Journey to the Dark Side.” Incorrectness Workshop at POPL’24, January 2024
2. “Under-approximate Reasoning at Scale.” *23rd Conference on High Confidence Software and Systems*, Annapolis, May 2023.
3. “Formal Reasoning and the Hacker Way.” Keynote. International Conference on Software Engineering, July 2020
4. “Formal Reasoning and the Hacker Way.” Keynote. International Workshop on the State Of the Art in Program Analysis, June 2020
5. “Continuous Reasoning: Scaling the Impact of Formal Methods.” Plenary invited lecture, *Federated Logic Conference*, Oxford, July 2018. (This was a joint meeting of 6 main conferences and 74 workshops)
6. “Experience Developing and Deploying Concurrency Analysis at Facebook.” Invited tutorial, *Static Analysis Symposium*, Freiburg, August 2018.
7. “From Categorical Logic to Facebook Engineering.” Joint Keynote Speaker for *Symposium on Logic in Computer Science* and *International Colloquium on Automata, Languages and Programming*, Kyoto, July 2015.
8. “Moving Fast with Software Verification.” Invited talk at *Computer Aided Verification*, July 2015, San Francisco.
9. “Deploying the Infer Program Analyzer at Facebook.” TAPAS, Edinburgh, September 2016
10. “Moving Fast with Software Verification.” APLAS, Pohang Korea, November 2015
11. “Moving Fast with Software Verification.” ICFEM, Newcastle, October 2015
12. “Moving Fast with Software Verification.” CAV, San Francisco, July 2015
13. “From Categorical Logic to Facebook Engineering.” LICS+ICALP joint invited talk, Kyoto, July 2015
14. “The Essence of Reynolds.” Tribute talk given at POPL, San Diego, January 2014.
15. “Program Logic and Analysis.” POPL Programming Languages Mentoring Workshop. San Diego, January 2014.
16. “The Essence of Reynolds.” Tribute talk given at Higher Order Programming with Effects workshop, Boston, September 2013.
17. “Program Logic and Analysis.” POPL Programming Languages Mentoring Workshop. Rome, January 2013.
18. “Algebraic Laws of Concurrency and Separation.” 13th International Conference on Relational and Algebraic Methods in Computer Science (RAMiCS 13). Cambridge UK. September 2012
19. “Separation Logic.” Invited tutorial, POPL 2012, Philadelphia.
20. “Algebra, Logic, Locality, Concurrency.” Joint invited lecture for 1st International Conference on Certified Programs and Proofs (CPP) and 7th Asian Symposium on Programming Languages and Systems (APLAS), Taiwan, December 2011.
21. “Reasoning About Programs using a Scientific Method.” 13th International Conference on Formal Engineering Methods, November 2011.
22. “Reasoning About Programs using a Scientific Method.” Joint London Mathematical Society/BCS-FACS Evening Seminar. November 2010.
23. “Abductive, Inductive and Deductive Reasoning about Resources.” *24th Annual Conference on Computer Science Logic*, Brno, August 2010
24. “Lectures on Separation Logic (4 lectures).” Featured Invited Lecturer, *Midlands Graduate School*, April 2010.

25. “Concurrent Separation Logic.” and “How to cook a static analyzer, or, the surprising effectiveness of substructural proof theory.” Invited Tutorials. *9th Conference on High Confidence Software and Systems*, Baltimore, May 2009.
26. “Proof Theory, Semantics and Operating Systems.” Keynote, *Midlands Graduate School Christmas Seminar*, December 2008.
27. “Separation Logic Tutorial.” *25th International Conference on Logic Programming*, Udine, December 2008.
28. ”Separation Logic and Concurrency.” 5 Lectures. *Laser Summer School on Concurrency and Correctness*, Elba, August 2008.
29. “Space Invading Systems Code.” Keynote lecture, at *18th International Symposium on Logic-Based Program Synthesis and Transformation*, Valencia, July 2008. (Plenary invited lecture for a meeting of 4 conferences.)
30. “Tutorial on Separation Logic.” Invited tutorial, at *30th International Conference on Computer Aided Verification*, Princeton, July 2008.
31. “Separation Logic Semantics of Communicating Processes” Keynote lecture, at *1st International Conference on Foundations of Informatics, Computing and Software*, Shanghai, June 2008.
32. “Concurrent Separation Logic.” Keynote lecture, at *Workshop on Verification of Concurrent Algorithms*, Cambridge, May 2008;
33. “Separation Logic and Concurrent Resource Management.” *6th International Symposium on Memory Management*, Montreal, October 2007;
34. “Separation Logic Semantics of Communicating Processes.” *18th Conference on Concurrency Theory*, Lisbon, September 2007;
35. “Proof Procedures for Separation Logic.” Invited lecture, at *Fifth International Workshop on Satisfiability Modulo Theories*, Berlin, July 2007;
36. “Separation Logic and Program Analysis” Invited lecture, at *13th International Static Analysis Symposium*, Seoul, August 2006.
37. “Concurrent Separation Logic.” Invited tutorial, at *22nd Conference on Mathematical Foundations of Program Semantics*, Genoa, May 2006.
38. “Local Action.” Keynote Lecture, at *6th Coalgebras Workshop*, Vienna, April 2006.
39. “Smallfoot: Automatic Modular Verification using Separation Logic.” Invited lecture, at *Formal Methods for Components and Objects*, Amsterdam, November 2005;
40. “Resources, Concurrency and Local Reasoning”, Invited tutorial (with S Brookes) at: *15th Conference on Concurrency Theory*, London, September 2004.
41. “Resources, Concurrency and Local Reasoning”, Plenary invited lecture at: *European Joint Conferences on Theory and Practice of Software*, Barcelona, April 2004.
42. “Separation and Information Hiding”, International Workshop on Aliasing, Confinement and Ownership in Object-oriented Programs, Darmstadt, July 2003.
43. “Local Reasoning about Programs that Alter Data Structures”, *Computer Science Logic*, Paris, September 2001.
44. “Local Reasoning about Shared Mutable Data Structure”, Workshop of EU Project: Applied Semantics. Darmstadt, Germany, March 2001.
45. “Reasoning about Shared Mutable Data Structure” (with J.C. Reynolds), SPACE: Symposium on Program Analysis and Computing Environments for Memory Management. London, January 2001.
46. “Semantics of Storage”, 16th annual symposium on Mathematical Foundations of Program Semantics, Hoboken, New Jersey, April 2000.
47. “Resource Interpretations of Bunched Implications,” Inaugural Workshop, Équipe Preuves, Programmes et Systèmes, CNRS, Université de Paris 7. 4-6 October 1999.

48. “Objects, Local State and Linear Polymorphism,” Workshop on Foundations of Object-Oriented Languages, San Diego, January, 1998
49. “From Algol to Polymorphic Linear Lambda-Calculus” Linear Logic ’96, Tokyo, March 1996.
50. “Kripke Logical Relations and PCF”, Conference on Logic, Domains and Programming Languages, Darmstadt, May 1995.
51. “Semantics of Storage”, Tutorial at ACM Workshop on State in Programming Languages, San Francisco, 1995.

Research Funding

1. Royal Society Wolfson Research Merit Award, 2007. These awards provide UK universities ”with additional financial support to attract key researchers to this country or to retain those who might seek to gain higher salaries elsewhere.” (<https://www.nature.com/articles/nj6871-03a>)
2. Royal Academy of Engineering/Microsoft Research Chair, 2012 (5-year research professorship, interrupted by move to Facebook in 2013)
3. UK EPSRC Programme Grant. Interface Reasoning for Interacting Systems. £6.1M. 1 January 2018 to 31 December 2023. (Prof DJ Pym is PI, I am coI with 6 others) According to EPSRC, ‘Programme grants are a flexible mechanism for providing funding to address significant major research challenges.’
4. EPSRC Programme Grant. Resource Reasoning. £3.2M. 1 January 2010 to 31 December 2015. (PI, with coPI’s Cook and Yang at QM; Gardner and Calcagno at IC). The PI position on this grant was assumed by Pym on 1/11/13 when I joined Facebook.
5. EPSRC Platform Grant. Extreme Reasoning. £720K. 1/12/2008 to 31/12/2011. (PI, with coPI’s Curzon, Honda, Oliva, Malacaria, Martin) In the words of EPSRC: ‘Platform Grants provide underpinning funding to world leading research groups.’
6. GCHQ Contract. Separation Logic Tools Support. £264,000. 1/5/2009 to 31/7/2012.
7. EPSRC Grant. Modularity and Resource Separation. £266,116. 01 October 2006 to 30 September 2009.
8. EPSRC Grant. Smallfoot: Static Assertion Checking for C programs. £324,918 01 August 2006 to 31 July 2009.
9. Microsoft PhD Studentship (in support of Aziem Chawdhary): Automatic Program Verification with Separation Logic. 1 Oct 2005 to 30 Sept 2008. £60,000.
10. EPSRC Visiting Fellowship Research Grant (in support of Jim Royer): Towards a Compositional Model of Complexity at Higher Types. £7050. 15 September to 14 December 2004.(with S Riis).
11. EPSRC Grant: Applications of Local Reasoning. £145K. 1 January, 2004 to 31 December, 2006. Research grant paired with advanced fellowship.
12. EPSRC Grant: Bunched ML (with EP Robinson). £167K. 1 November, 2003 to 31 October, 2006. The Bath site of this grant, under David Pym, has a further £152K.
13. EPSRC Advanced Fellowship: Local Reasoning: Foundations and Applications. £250K. 1 October, 2003 to 31 September, 2008.
14. Nuffield Foundation Summer Research Fellowship, in support of A. Chawdhary. £1.5K. Summer, 2003.
15. ESPRIT Working Group: APPSEM II. 60,000euros (approx). 2003-2006.
16. EPSRC Visiting Fellowship Research Grant (in support of John Reynolds): Concurrency and Code Pointers in Spatial Pointer Logic. £46,077. 15 July 2002 to 15 Jan 2003.

17. EPSRC Visiting Fellowship Research Grant (in support of A. Banerjee, D. Naumann and H. Yang): Abstraction, Confinement and Heap Storage. £8,850. 1 August 2002 to 31 September 2002.
18. EPSRC Grant: Local Reasoning about State (with D. Pym and R. Bornat). £149K. 1 March 2001 to 28 Feb 2004
19. EPSRC grant: Verified Bytecode (with R. Bornat). £169,196. 1 Dec, 1997 to 30 Nov, 2000.
20. EPSRC Visiting Fellowship Research Grant (in support of U.S. Reddy): Parametricity and Reflexive Graphs. £4,750. 1 May, 1998 to 31 July, 1998.
21. EPSRC grant: Logic Programming, Imperative Programming and Categorical Semantics (with E. Robinson and D. Pym). £149,500. 21 July, 1997 to 20 July, 2000.
22. EPSRC Visiting Fellowship Research Grant (in support of J.C. Reynolds): Types, State and Polymorphism. £6,783. 1 August, 1997 to 31 July, 1998.
23. National Science Foundation (USA). Research initiation award. \$90,000. Awarded May 1992. Period: June 1, 1992 to Dec 31, 1995.
24. Natural Sciences and Engineering Council of Canada Postgraduate Scholarship, 1987-1989

Teaching and Academic Administration

Departmental Responsibilities at UCL

- Founder and Head of PPLV Research Group, 2012–2013
- Deputy coordinator for Research Excellence Framework submission, 2012-2013

Departmental Responsibilities at Queen Mary

- Seminar Organiser (1997 and 98/99)
- BSc Teaching Committee (1998/99)
- BSc Project Co-ordinator (1997/98 and 98/99)
- MSc IT Project Co-ordinator (2000/01)
- Chair of Taught Courses Committee (2000-03)
- Review of BSc Curriculum (2003-03)
- Head of Theoretical Computer Science Research Group (2003-2012)

Teaching at Queen Mary

- Master’s course in Program Semantics. Autumn 97 and 98
- Compilers and Interpreters. Spring 97, 99, Winter 00, Autumn 01.
- Computability. Fall 98 (1/3)
- Computer Systems 3. Spring 98 (1/3) and 99 (1/3)
- Introduction to Algorithms. Spring 98 (1/2)
- Introduction to Programming. Fall 98 (1/4), Autumn 98, Fall 99, Fall/Spring 2000.
- Foundations 3, Autumn 2002.
- Language and Communication), Winter 2002, 03, 09.

Teaching at Syracuse

- Numerous courses at Syracuse University, 1990-1995, from 1st year through to PhD level. 4 years teaching Introduction to Programming. Introduced Scheme programming into first year.
- Instituted course on mobile robotics.
- Redesigned first year curriculum for CS majors; designed (with 2 others) curriculum for service courses (for non-CS majors).

PhD Students and RAs

Steve Cooper. PhD 1997, Syracuse University. (Now Associate Professor at Stanford)

Paul Levy. PhD 2001, Queen Mary. (Now Lecturer and EPSRC Advanced Fellow at Birmingham)

Cristiano Calcagno. PhD 2002 (jointly supervised by E Moggi and I). RA 2002-2004. (Codounded Monoidics, worked at Facebook, now advising startup companioes)

Josh Berdine, PhD 2003, Queen Mary. RA from 2003-2005. (Now at Facebook)

Ivana Mijajlović, PhD 2007, Queen Mary. RA from 2007-2009.

Akbar Hussain, PhD 2013

Stefano Guerrini, RA 1997-1998. (Now Lecturer at Università di Roma)

Hayo Thielecke, RA 1997-2000. (Now a Lecturer at Birmingham)

Samin Ishtiaq, RA 1998-2000. (Now at Microsoft Research Cambridge)

Dino Distefano, RA 2004-2007. (Cofounded Monoidics, now at Facebook)

Rasmus Petersen, RA, 2007-2009. (Now Postdoc at Microsoft Research Cambridge.)

Noam Rinetzky, RA 2008-2010. (Now Associate Professor at Tel Aviv University)

Nikos Gorogiannis, RA 2010-2012. (Now at Facebook)

Jules Villard, RA 2010-2013. (Now at Facebook)

Quang Loc Le, RA 2020-2022 . (Now Lecturer at UCL)

Education

PhD in Computer Science, Queen’s University, Kingston, Canada, 1991. (supervisor: R. D. Tennent)

MSc in Computer Science, Queen’s 1987. (supervisor: Z. Stachniak)

BSc in Computer Science, Dalhousie University, 1985.

Personal Information

Born in Halifax, Nova Scotia, 13 July 1963. British and Canadian citizen.
Residence: London, UK