Bibliography

[1] Rodney M. Burstall. Some techniques for proving correctness of programs which alter data structures. In Bernard Meltzer and Donald Michie, editors, *Machine Intelligence 7*, pages 23–50. Edinburgh University Press, Edinburgh, Scotland, 1972.

[2] Tomasz Kowaltowski. *Correctness of Programs Manipulating Data Structures*. Ph. D. dissertation, University of California, Berkeley, California, December 1973. Also Electronics Research Laboratory Memorandum ERL-M404, September 1973.

[3] Stephen A. Cook and Derek C. Oppen. An assertion language for data structures. In *Conference Record of the Second ACM Symposium on Principles of Programming Languages*, pages 160–166, New York, 1975. ACM.

[4] Derek C. Oppen and Stephen A. Cook. Proving assertions about programs that manipulate data structures. In *Proceedings of Seventh Annual ACM Symposium on Theory of Computing*, pages 107–116, New York, 1975. ACM.

[5] Joseph M. Morris. A general axiom of assignment; assignment and linked data structures; a proof of the Schorr-Waite algorithm. In Manfried Broy and Gunther Schmidt, editors, *Theoretical Foundations of Programming Methodology*, pages 25–51. D. Reidel, Dordrecht, Holland, 1982.

[6] Ian A. Mason. *The Semantics of Destructive Lisp*. CSLI Lecture Notes Number 5. Center for the Study of Language and Information, Menlo Park, CA, 1986.

[7] Ian A. Mason. Verification of programs that destructively manipulate data. *Science of Computer Programming*, 10(2):177–210, April 1988.

[8] Ian A. Mason and Carolyn Talcott. Equivalence in functional languages with effects. *Journal of Functional Programming*, 1(3):287–327, July 1991.

[9] Furio Honsell, Ian A. Mason, Scott Smith, and Carolyn Talcott. A variable typed logic of effects. *Information and Computation*, 119(1):55–90, May 15, 1995.

[10] Ian A. Mason. A first order logic of effects. *Theoretical Computer Science*, 185(2):277–318, October 20, 1997.

[11] Andrew M. Pitts and Ian D. B. Stark. Operational reasoning for functions with local state. In Andrew D. Gordon and Andrew M. Pitts, editors, *Higher Order Operational Techniques in Semantics*, pages 227–273. Cambridge University Press, 1998.

[12] Ian D. B. Stark. *Names and Higher-Order Functions*. Ph. D. dissertation, University of Cambridge, Cambridge, England, December 1994.

[13] Ian D. B. Stark. Categorical models for local names. *Lisp and Symbolic Computation*, 9(1):77–107, February 1996.

[14] Ian D. B. Stark. Names, equations, relations: Practical ways to reason about *new*. *Fundamenta Informaticae*, 33:369–396, 1998.

[15] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580 and 583, October 1969.

[16] C. A. R. Hoare. Proof of a program: FIND. *Communications of the ACM*, 14(1):39–45, January 1971.

[17] Robert W. Floyd. Assigning meanings to programs. In J. T. Schwartz, editor, *Mathematical Aspects of Computer Science*, volume 19 of *Proceedings of Symposia in Applied Mathematics*, pages 19–32, Providence, Rhode Island, 1967. American Mathematical Society.

[18] Peter Naur. Proof of algorithms by general snapshots. *BIT*, 6:310–316, 1966.

[19] D. I. Good. *Towards a Man-machine System for Proving Program Correctness*. Ph. D. dissertation, University of Wisconsin, Madison, Wisconsin, 1970.

[20] Samin Ishtiaq and Peter W. O'Hearn. BI as an assertion language for mutable data structures. In *Conference Record of POPL 2001: The 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 14–26, New York, 2001. ACM.

[21] John C. Reynolds. Intuitionistic reasoning about shared mutable data structure. In Jim Davies, Bill Roscoe, and Jim Woodcock, editors, *Millennial Perspectives in Computer Science*, pages 303–321, Houndsmill, Hampshire, 2000. Palgrave.

[22] Peter W. O'Hearn and David J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, June 1999.

[23] David J. Pym. *The Semantics and Proof Theory of the Logic of Bunched Implications*. Applied Logic Series. Kluwer Academic Publishers, Boston, Massachusetts, 2002. (to appear).

[24] Peter W. O'Hearn, John C. Reynolds, and Hongseok Yang. Local reasoning about programs that alter data structures. In Laurent Fribourg, editor, *Computer Science Logic*, volume 2142 of *Lecture Notes in Computer Science*, pages 1–19, Berlin, 2001. Springer-Verlag.

[25] Hongseok Yang and Peter W. O'Hearn. A semantic basis for local reasoning. In M. Nielsen and U. Engberg, editors, *Foundations of Software Science and Computation Structures*, volume 2303 of *Lecture Notes in Computer Science*, pages 402–416, Berlin, 2002. Springer-Verlag.

[26] John C. Reynolds and Peter W. O'Hearn. Reasoning about shared mutable data structure (abstract of invited lecture). In Fritz Henglein, John Hughes, Henning Makholm, and Henning Niss, editors, *SPACE 2001: Informal Proceedings of Workshop on Semantics, Program Analysis and Computing Environments for Memory Management*, page 7. IT University of Copenhagen, 2001. The slides for this lecture are available at ftp://ftp.cs.cmu.edu/user/jcr/spacetalk.ps.gz.

[27] H. Schorr and William M. Waite. An efficient machine-independent procedure for garbage collection in various list structures. *Communications of the ACM*, 10:501–506, 1967.

[28] Hongseok Yang. An example of local reasoning in BI pointer logic: The Schorr-Waite graph marking algorithm. In Fritz Henglein, John Hughes, Henning Makholm, and Henning Niss, editors, *SPACE 2001: Informal Proceedings of Workshop on Semantics, Program Analysis and Computing Environments for Memory Management*, pages 41–68. IT University of Copenhagen, 2001.

[29] Hongseok Yang. *Local Reasoning for Stateful Programs*. Ph. D. dissertation, University of Illinois, Urbana-Champaign, Illinois, July 2001.

[30] C. J. Cheney. A nonrecursive list compacting algorithm. *Communications of the ACM*, 13(11):677–678, November 1970.

[31] Noah Torp-Smith, Lars Birkedal, and John C. Reynolds. Local reasoning about a copying garbage collector. *ACM Transactions on Programming Languages and Systems*, 30(4):24:1–58, 2008.

[32] Carsten Varming and Lars Birkedal. Higher-order separation logic in Isabelle/HOLCF. To appear in the Proceedings of the 24th Annual Conference on Mathematical Foundations of Programming Semantics, Electronic Notes in Theoretical Computer Science, http://www.elsevier.{nl,com}/locate/entcs, 2008.

[33] Cristiano Calcagno, Hongseok Yang, and Peter W. O'Hearn. Computability and complexity results for a spatial assertion language for data structures. In Ramesh Hariharan, Madhavan Mukund, and V. Vinay, editors, *FST TCS 2001: Foundations of Software Technology and Theoretical Computer Science*, volume 2245 of *Lecture Notes in Computer Science*, pages 108–119, Berlin, 2001. Springer-Verlag.

[34] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proceedings Seventeenth Annual IEEE Symposium on Logic in Computer Science*, pages 55–74, Los Alamitos, California, 2002. IEEE Computer Society.

[35] Peter W. O'Hearn, Hongseok Yang, and John C. Reynolds. Separation and information hiding. In *Conference Record of POPL 2004: The 31st*

ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pages 268–280, New York, 2004. ACM Press.

[36] Peter W. O'Hearn, Hongseok Yang, and John C. Reynolds. Separation and information hiding. To appear in the ACM Transactions on Programming Languages and Systems, 2009.

[37] Aleksandar Nanevski, Greg Morrisett, and Lars Birkedal. Hoare type theory, polymorphism and separation. To appear in the the Journal of Functional Programming, 2007.

[38] Kasper Svendsen, Alexandre Buisse, and Lars Birkedal. Verifying design patterns in Hoare type theory. Report TR-2008-112, IT University of Copenhagen, Copenhagen, Denmark, October 2008.

[39] Rasmus Lerchedahl Petersen, Lars Birkedal, Aleksandar Nanevski, and Greg Morrisett. A realizability model of impredicative Hoare type theory. In *Proceedings of ESOP 2008*, 2008.

[40] Aleksandar Nanevski, Greg Morrisett, and Lars Birkedal. Polymorphism and separation in Hoare type theory. In *ICFP '06: Proceedings of the Eleventh ACM SIGPLAN International Conference on Functional Programming*, pages 62–73, New York, 2006. ACM.

[41] Lars Birkedal, Noah Torp-Smith, and Hongseok Yang. Semantics of separation-logic typing and higher-order frame rules for Algol-like languages. *Logical Methods in Computer Science*, 2(5:1), August 2006.

[42] Lars Birkedal and Hongseok Yang. Relational parametricity and separation logic. *Logical Methods in Computer Science*, 4(2:6):1–27, 2008.

[43] Hongseok Yang. Relational separation logic. *Theoretical Computer Science*, 375(1–3):308–334, May 2007.

[44] John C. Reynolds. Types, abstraction and parametric polymorphism. In R. E. A. Mason, editor, *Information Processing 83*, pages 513–523, Amsterdam, 1983. Elsevier Science Publishers B. V. (North-Holland).

[45] P. Nick Benton and B. Leperchey. Relational reasoning in a nominal semantics for storage. In *Proceedings of the Seventh International Conference on Typed Lambda Calculi and Applications (TLCA'05)*, volume

3461 of *Lecture Notes in Computer Science*, Berlin, 2005. Springer-Verlag.

[46] Bodil Biering, Lars Birkedal, and Noah Torp-Smith. Bi-hyperdoctrines, higher-order separation logic, and abstraction. *ACM Transactions on Programming Languages and Systems*, 29(5):24:1–35, 2007.

[47] Bernard Reus and J. Schwinghammer. Separation logic for higher-order state. In *Proceedings CSL'06*, volume 4207 of *Lecture Notes in Computer Science*, pages 575–590, Berlin, 2006. Springer-Verlag.

[48] Lars Birkedal, Bernard Reus, J. Schwinghammer, and Hongseok Yang. A simple model of separation logic for higher-order store. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, volume 5126 of *Lecture Notes in Computer Science*, pages 348–360, Berlin, July 2008. Springer-Verlag.

[49] Ivana Mijajlović and Noah Torp-Smith. Refinement in a separation context. In *SPACE 2004: Informal Proceedings of Second Workshop on Semantics, Program Analysis and Computing Environments for Memory Management*, 2004.

[50] Ivana Mijajlović, Noah Torp-Smith, and Peter W. O'Hearn. Refinement and separation contexts. In Kamal Lodaya and Meena Mahajan, editors, *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science*, volume 3328 of *Lecture Notes in Computer Science*, pages 421–433, Berlin, 2004. Springer-Verlag.

[51] Ivana Mijajlović and Hongseok Yang. Data refinement with low-level pointer operations. In *Proceedings of the 3rd Asian Symposium on Programming Languages*, volume 3780 of *Lecture Notes in Computer Science*, pages 19–36, Berlin, 2005. Springer-Verlag.

[52] Matthew J. Parkinson and Gavin Bierman. Separation logic and abstraction. In *Conference Record of POPL 2005: The 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 247–258, New York, 2005. ACM Press.

[53] Matthew J. Parkinson. *Local Reasoning in Java*. Ph. D. dissertation, University of Cambridge, Cambridge, England, August 2005.

[54] Matthew J. Parkinson and Gavin Bierman. Separation logic, abstraction, and inheritance. In *Conference Record of POPL 2008: The 35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, New York, 2008. ACM Press.

[55] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley, Reading, Massachusetts, 1995.

[56] Neelakantan R. Krishnaswami. Reasoning about iterators with separation logic. In *SAVCBS '06: Proceedings of the 2006 Conference on Specification and Verification of Component-Based Systems*, pages 83–86, New York, 2006. ACM.

[57] Neelakantan R. Krishnaswami, Lars Birkedal, and Jonathan Aldrich. Modular verification of the subject-observer pattern via higher-order separation logic. Unpublished draft, presented at the FTFJP 2007 Workshop, 2007.

[58] Neelakantan R. Krishnaswami, Jonathan Aldrich, Lars Birkedal, Kasper Svendsen, and Alexandre Buisse. Design patterns in separation logic. To appear in Types for Language Design and Implementation (TLDI), 2009.

[59] Peter W. O'Hearn. Resources, concurrency and local reasoning. In *CONCUR 2004 — Concurrency Theory, Proceedings of the 15th International Conference*, volume 3170 of *Lecture Notes in Computer Science*, pages 49–67, Berlin, 2004. Springer-Verlag.

[60] Stephen D. Brookes. A semantics for concurrent separation logic. In *CONCUR 2004 — Concurrency Theory, Proceedings of the 15th International Conference*, volume 3170 of *Lecture Notes in Computer Science*, pages 16–34, Berlin, 2004. Springer-Verlag.

[61] Matthew J. Parkinson, Richard Bornat, and Peter W. O'Hearn. Modular verification of a non-blocking stack. In *Conference Record of POPL 2007: The 34th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, New York, 2007. ACM Press.

[62] Cliff B. Jones. Specification and design of (parallel) programs. In R. E. A. Mason, editor, *Information Processing 83*, Amsterdam, 1983. Elsevier Science Publishers B. V. (North-Holland).

[63] Viktor Vafeiadis and Matthew J. Parkinson. A marriage of rely/guarantee and separation logic. In *CONCUR 2007 — Concurrency Theory, Proceedings of the 18th International Conference*, Lecture Notes in Computer Science, Berlin, 2007. Springer-Verlag.

[64] Cristiano Calcagno, Viktor Vafeiadis, and Matthew J. Parkinson. Modular safety checking for fine-grained concurrency. In *Proceedings SAS 2007*, 2007.

[65] Xinyu Feng. Local rely-guarantee reasoning. To appear in Conference Record of POPL 2009: The 36th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, 2009.

[66] J. Hayman and Glynn Winskel. Independence and concurrent separation logic. In *Proceedings 21st Annual IEEE Symposium on Logic in Computer Science*, pages 147–156, Los Alamitos, California, 2006. IEEE Computer Society Press.

[67] Harvey Tuch and Gerwin Klein. A unified memory model for pointers. In G. Sutcliffe and A. Voronkov, editors, *12th International Conference on Logic for Programming Artificial Intelligence and Reasoning*, volume 3835 of *Lecture Notes in Computer Science*, pages 474–488, Berlin, 2005. Springer-Verlag.

[68] Harvey Tuch, Gerwin Klein, and Michael Norrish. Types, bytes, and separation logic. In *Conference Record of POPL 2007: The 34th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, New York, 2007. ACM Press.

[69] Dachuan Yu, Nadeem A. Hamid, and Zhong Shao. Building certified libraries for PCC: Dynamic storage allocation. *Science of Computer Programming*, 50:101–127, 2004.

[70] Cristiano Calcagno, Josh Berdine, Hongseok Yang, and Peter W. O'Hearn. Saw: The spatial assertion workbench. In *Proceedings of the 2nd Workshop on Automated Verification of Critical Systems*, 2002.

[71] Josh Berdine, Cristiano Calcagno, and Peter W. O'Hearn. A decidable fragment of separation logic. In Kamal Lodaya and Meena Mahajan, editors, *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science*, volume 3328 of *Lecture Notes in Computer Science*, pages 97–109, Berlin, 2004. Springer-Verlag.

[72] Oukseh Lee, Hongseok Yang, and Kwangkeun Yi. Automatic verification of pointer programs using grammar-based shape analysis. In *Proceedings of the 14th European Symposium on Programming*, volume 3444 of *Lecture Notes in Computer Science*, pages 124–140, Berlin, 2005. Springer-Verlag.

[73] Dino Distefano, Peter W. O'Hearn, and Hongseok Yang. A local shape anaylsis based on separation logic. In *Proceedings of the 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 3920 of *Lecture Notes in Computer Science*, pages 287–302, Berlin, 2006. Springer-Verlag.

[74] Cristiano Calcagno, Dino Distefano, Hongseok Yang, and Peter W. O'Hearn. Beyond reachability: Shape abstraction in the presence of pointer arithmetic. In *Proceedings of the 13th International Static Analysis Symposium*, volume 4134 of *Lecture Notes in Computer Science*, pages 182–203, Berlin, 2006. Springer-Verlag.

[75] Cristiano Calcagno, Dino Distefano, Peter W. O'Hearn, and Hongseok Yang. Footprint analysis: A shape analysis that discovers preconditions. In *Proceedings of the 14th International Static Analysis Symposium*, volume 4634 of *Lecture Notes in Computer Science*, pages 402–418, Berlin, 2007. Springer-Verlag.

[76] Josh Berdine, Cristiano Calcagno, Byron Cook, Dino Distefano, Peter W. O'Hearn, Thomas Wies, and Hongseok Yang. Shape analysis for composite data structures. In *Proceedings of the 19th International Conference on Computer Aided Verification*, volume 4590 of *Lecture Notes in Computer Science*, pages 178–192, Berlin, 2007. Springer-Verlag.

[77] Hongseok Yang, Oukseh Lee, Josh Berdine, Cristiano Calcagno, Byron Cook, Dino Distefano, and Peter W. O'Hearn. Scalable shape analysis for systems code. In *Proceedings of the 20th International Conference*

on Computer Aided Verification, volume 5123 of *Lecture Notes in Computer Science*, pages 385–398, Berlin, 2008. Springer-Verlag.

[78] John Boyland. Checking interference with fractional permissions. In Radhia Cousot, editor, *Static Analysis: 10th International Symposium*, volume 2694 of *Lecture Notes in Computer Science*, pages 55–72, Berlin, 2003. Springer-Verlag.

[79] Richard Bornat, Cristiano Calcagno, Peter W. O'Hearn, and Matthew Parkinson. Permission accounting in separation logic. In *Conference Record of POPL 2005: The 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 259–270, New York, 2005. ACM Press.

[80] Richard Bornat, Cristiano Calcagno, and Hongseok Yang. Variables as resource in separation logic. In *Proceedings of the 21st Annual Conference on Mathematical Foundations of Programming Semantics*, volume 155 of *Electronic Notes in Theoretical Computer Science*, pages 247–276, May 2006.

[81] Matthew J. Parkinson, Richard Bornat, and Cristiano Calcagno. Variables as resource in Hoare logic. In *Proceedings 21st Annual IEEE Symposium on Logic in Computer Science*, Los Alamitos, California, 2006. IEEE Computer Society Press.

[82] Cristiano Calcagno, Peter W. O'Hearn, and Hongseok Yang. Local action and abstract separation logic. In *Proceedings 22nd Annual IEEE Symposium on Logic in Computer Science*, pages 366–378, Los Alamitos, California, July 2007. IEEE Computer Society.

[83] Lawrence C. Paulson. Isabelle: The next 700 theorem provers. In P. Odifreddi, editor, *Logic and Computer Science*, pages 361–386. Academic Press, 1990.

[84] Michael J. Gordon. Introduction to the HOL system. In *International Workshop on the HOL Theorem Proving System*, pages 2–3, 1991.

[85] T. Weber. Towards mechanized program verification with separation logic. In J. Marcinkowski and A. Tarlecki, editors, *Computer Science Logic — 18th International Workshop*, volume 3210 of *Lecture Notes in Computer Science*, pages 250–264, Berlin, 2004. Springer-Verlag.

[86] N. Schirmer. *Verification of Sequential Imperative Programs in Isabelle/HOL*. Ph. D. dissertation, Technische Universität München, Munich, Germany, 2006.

[87] Luis Caires and L. Monteiro. Verifiable and executable specifications of concurrent objects in $\mathcal{L}_\pi$. In C. Hankin, editor, *Programming Languages and Systems — ESOP '98*, volume 1381 of *Lecture Notes in Computer Science*, pages 42–56, Berlin, 1998. Springer-Verlag.

[88] Luca Cardelli and Andrew D. Gordon. Anytime, anywhere: Modal logics for mobile ambients. In *Conference Record of POPL '00: The 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 365–377, New York, 2000. ACM.

[89] G. Conforti, Giorgio Ghelli, A. Albano, D. Colazzo, P. Manghi, and C. Sartiani. The query language TQL. In *Proceedings of the 5th International Workshop on the Web and Databases (WebDB)*, Madison, Wisconsin, 2002.

[90] Luca Cardelli, Philippa Gardner, and Giorgio Ghelli. A spatial logic for querying graphs. In Matthew Hennessy and P. Widmayer, editors, *Automata, Languages and Programming*, Lecture Notes in Computer Science, Berlin, 2002. Springer-Verlag.

[91] Luca Cardelli and Giorgio Ghelli. A query language based on the ambient logic. In D. Sands, editor, *Programming Languages and Systems — ESOP 2001*, volume 2028 of *Lecture Notes in Computer Science*, pages 1–22, Berlin, 2001. Springer-Verlag.

[92] Cristiano Calcagno, Philippa Gardner, and Uri Zarfaty. Context logic and tree update. In *Conference Record of POPL 2005: The 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 271–282, New York, 2005. ACM Press.

[93] John C. Reynolds. *The Craft of Programming*. Prentice-Hall International, London, 1981.

[94] C. A. R. Hoare. Towards a theory of parallel programming. In C. A. R. Hoare and R. H. Perrott, editors, *Operating Systems Techniques*, volume 9 of *A.P.I.C. Studies in Data Processing*, pages 61–71, London, 1972. Academic Press.

[95] Susan Speer Owicki and David Gries. Verifying properties of parallel programs: An axiomatic approach. *Communications of the ACM*, 19(5):279–285, May 1976.

[96] Stephen Cole Kleene. *Introduction to Metamathematics*, volume 1 of *Bibliotheca Mathematica*. North-Holland, Amsterdam, 1952.

[97] John C. Reynolds. *Theories of Programming Languages*. Cambridge University Press, Cambridge, England, 1998.

[98] Jacques Loeckx, Kurt Sieber, and Ryan D. Stansifer. *The Foundations of Program Verification*. Wiley, Chichester, England, second edition, 1987.

[99] John McCarthy. Recursive functions of symbolic expressions and their computation by machine, part I. *Communications of the ACM*, 3(4):184–195, April 1960.

[100] C. A. R. Hoare. Algorithm 63: Partition. *Communications of the ACM*, 4(7):321, July 1961.

[101] C. A. R. Hoare. Algorithm 64: Quicksort. *Communications of the ACM*, 4(7):321, July 1961.