

David A. Naumann

September 6, 2008

Professor of Computer Science, Stevens Institute of Technology, Hoboken NJ 07030.

URL: <http://www.cs.stevens.edu/~naumann/>

Education

University of Texas at Austin, Computer Science, BA 1982

University of Texas at Austin, Computer Science, PhD 1992

Appointments

Professor of Computer Science, Stevens Institute of Technology, since 2008

Associate Professor of Computer Science, Stevens Institute of Technology, 2002–08

Assistant Professor of Computer Science, Stevens Institute of Technology, 1997–2002

Assistant Professor of Mathematics and Computer Science, Southwestern University, 1991–97

Associate Scientist, International Software Systems, Austin, 1986–91

Consultant-programmer, Renaissance Systems, Austin, 1985–86

Programmer-designer, IBM, Austin, 1982–85

Journal Papers

1. A recursion theorem for predicate transformers on inductive data types. *Information Processing Letters* 50 (1994) 329–336.
2. Data refinement, call by value, and higher order programs. *Formal Aspects of Computing* 7 (1995) 652–662.
3. Predicate transformers and higher order programs. *Theoretical Computer Science* 150 (1995) 111–159.
4. A categorical model of higher order imperative programming. *Mathematical Structures in Computer Science* 8 (1998) 351–399.
5. A weakest precondition semantics for refinement of object-oriented programs (with Ana Cavalcanti). *IEEE Transactions on Software Engineering* 26 (2000) 713–728.
6. Calculating sharp adaptation rules. *Information Processing Letters* 77 (2001) 201–208.
7. Predicate transformer semantics of a higher order imperative language with record subtyping. *Science of Computer Programming* 41 (2001) 1–51.
8. Soundness of data refinement for a higher order imperative language. *Theoretical Computer Science* 278 (2002) 271–301.

9. Stack-based access control for secure information flow (with Anindya Banerjee). *Journal of Functional Programming* 15 (2005) 131–177.
10. Ownership confinement ensures representation independence for object-oriented programs (with Anindya Banerjee). *Journal of the ACM* 52 (2005) 894–960.
11. Towards imperative modules: Reasoning about invariants and sharing of mutable state (with Michael Barnett). *Theoretical Computer Science* 365 (2006) 143–168.
12. Observational purity and encapsulation. *Theoretical Computer Science* 376 (2007) 205–224.
13. On assertion-based encapsulation for object invariants and simulations. *Formal Aspects of Computing* 19 (2007) 205–224. (Coincidentally, same pages as previous item.)

Proceedings of Refereed Conferences

§ indicates co-authors who were students at the time of submission.

1. Derivation of programs for freshmen (with R. Denman, W. Potter, and G. Richter). *ACM Conference of the Special Interest Group in Computer Science Education SIGCSE* (1994) 116–120.
2. Predicate transformer semantics of an Oberon-like language. *IFIP TC2 Working Conference on Programming Concepts, Methods and Calculi* (1994) 467–487.
3. On the essence of Oberon. *International Conference on Programming Languages and System Architecture*, Zürich, Jürg Gutknecht, ed. (1994) 313–327.
4. Beyond Fun: order and membership in polytypic imperative programming. *5th International Conference on Mathematics of Program Construction*, LNCS 1422 (1998) 286–314.
5. Towards squiggly refinement algebra. *Programming Concepts and Methods* (1998) 346–365.
6. A weakest precondition semantics for refinement in an object-oriented language (with Ana Cavalcanti). *World Congress on Formal Methods* (1999) 1439–1459.
7. Ideal models for pointwise relational and state-free imperative programming. *ACM Principles and Practice of Declarative Programming* (2001) 4–15.
8. Representation independence, confinement, and access control (with Anindya Banerjee). *29th ACM Symposium on Principles of Programming Languages* (2002) 166–177.
9. Forward simulation for data refinement of classes (with Ana Cavalcanti). *International Symposium of Formal Methods Europe* (2002) 471–490.

10. On a specification-oriented model for object-orientation (with Ana Cavalcanti). *6th Brazilian Symposium on Programming Languages* (2002) 114–127.
11. Secure information flow and pointer confinement in a Java-like language (with Anindya Banerjee). *15th IEEE Computer Security Foundations Workshop*¹ (2002) 253–270.
12. Using access control for secure information flow in a Java-like language (with Anindya Banerjee). *16th IEEE Computer Security Foundations Workshop* (2003) 155–169.
13. CodeBLUE: a Bluetooth interactive dance club system (with Dennis Hromin[§], Michael Chladil[§], Natalie Vanatta[§], Susanne Wetzel, Farooq Anjum, and Ravi Jain). *IEEE Global Telecommunications Conference* (2003) 2814–2818.
14. Towards imperative modules: Reasoning about invariants and sharing of mutable state, extended abstract (with Mike Barnett). *19th IEEE Symposium on Logic in Computer Science* (2004) 313–323.
15. Friends need a bit more: Maintaining invariants over shared state (with Mike Barnett) *7th International Conference on Mathematics of Program Construction*, LNCS 3125 (2004) 54–84.
16. Modular and constraint-based information flow inference for an object-oriented language (with Qi Sun[§] and Anindya Banerjee). *11th International Static Analysis Symposium*, LNCS 3148 (2004) 84–99.
17. Assertion-based encapsulation, object invariants and simulations (invited survey paper), in post-proceedings, *Formal Methods for Components and Objects*, LNCS 3657 (2004) 251–273.
18. Observational purity and encapsulation. *Fundamental Aspects of Software Engineering* (2005) 190–204. Awarded Best Software Sciences Paper of ETAPS 2005.
19. State based ownership, reentrance, and encapsulation (with Anindya Banerjee). *19th European Conference on Object-Oriented Programming* (2005) 387–411.
20. Verifying a secure information flow analyzer. *18th International Conference on Theorem Proving in Higher Order Logics* (2005) 211–226.
21. Deriving an information flow checker and certifying compiler for Java (with Gilles Barthe and Tamara Rezk[§]). *27th IEEE Symposium on Security and Privacy* (2006) 230–242.
22. From coupling relations to mated invariants for checking secure information flow. *11th European Symposium on Research in Computer Security* (2006) 279–296.

¹Nominally this was a refereed workshop, not a conference, but at the quality standard of a symposium—indeed, in 2007 it was renamed to the *20th Computer Security Foundations Symposium*.

23. Closing internal timing channels by transformation (with Alejandro Russo[§], John Hughes, and Andrei Sabelfeld). In *11th Asian Computing Science Conference* (2006).
24. Allowing state changes in specifications (with Michael Barnett, Wolfram Schulte, and Qi Sun[§]). *International Conference on Emerging Trends in Information and Communication Security* (2006) 321–336, invited paper.
25. Category theoretic models of data refinement (with Michael Johnson and John Power). In *Irish Conference on Mathematical Foundations of Computer Science and Information Technology* (2006), invited paper (proceedings to appear in 2007).
26. Beyond stack inspection: a unified access-control and information-flow security model (with Marco Pistoia and Anindya Banerjee). In *28th IEEE Symposium on Security and Privacy* (2007) 149–163.
27. Modular verification of higher-order methods with mandatory calls specified by model programs (with Steve M. Shaner[§] and Gary T. Leavens). In *22d International Conference on Object Oriented Programming, Languages, and Systems* (2007) 351–368. Awarded Best Student Paper.
28. Expressive declassification policies and modular static enforcement (with Anindya Banerjee and Stan Rosenberg[§]). In *29th IEEE Symposium on Security and Privacy* (2008) 339–353.
29. Regional logic for local reasoning about global invariants (with Anindya Banerjee and Stan Rosenberg[§]). in *European Conference on Object Oriented Programming* 2008. Awarded Distinguished Paper.
30. Boogie meets regions: a verification experience report (with Anindya Banerjee and Mike Barnett). In *Verified Software: Theories, Tools, Experiments* 2008, to appear.

Selected awards

Best Software Science Paper, ETAPS 2005 (European Association of Software Sciences and Technology).

Davis Memorial Award for Research Excellence, Stevens Institute of Technology, 2006.

Best Student Paper, OOPSLA 2008 (well, the student is Steven M. Shaner and the third coauthor is his advisor, Gary T. Leavens).

Distinguished Paper Award, ECOOP 2008 (with coauthors Anindya Banerjee and Stan Rosenberg).

Selected activities

Co-organizer and co-editor of proceedings, first Dagstuhl seminar on Language Based Security (Seminar 03411, October 5–10, 2003, <http://www.dagstuhl.de/03411/>) with Anindya Banerjee, Heiko Mantel, and Andrei Sabelfeld.

Co-editor of *Concurrency and Computation: Practice and Experience*, June 2004, special issue on formal methods for Java programs.

Co-author, *Verified Software Initiative: grand challenge roadmap 2007*.

Chair of Theory Working Group, Verified Software Initiative,² 2005–2007

Co-chair of Theory Workshop in connection with International Conference on *Verified Software: Theory, Tools, Experiments* 2008 (with Peter O’Hearn).

Co-chair of 4th ACM Workshop on Programming Languages and Analysis for Security in connection with International Conference on *Programming Language Design and Implementation* 2008 (with Stephen Chong).

Corresponding member, Verified Software Repository Network (<http://vsr.sourceforge.net/>)

Program committee member, conferences:

Formal Methods for Open Object-based Distributed Systems (FMOODS) 2007, 2008;

Verified Software: Theories, Tools, Experiments (VSTTE) 2008;

TOOLS Europe 2008;

17th European Symposium on Programming (ESOP) 2004, 2008;

European Symposium on Research in Computer Security (ESORICS) 2007;

14th International Symposium on Formal Methods (FM) 2006;

International Conference on Mathematics of Program Construction 2000, 2002, 2004;

Brazilian Symposium on Formal Methods 2004, 2007, 2008;

Brazilian Symposium on Programming Languages 2004, 2005, 2006, 2007, 2008;

Brazilian Symposium on Software Engineering 2002, 2003, 2005;

International Conference on Formal Engineering Methods 2005.

International Symposium on Unifying Theories of Programming (UTP) 2006, 2008.

Program committee member, refereed workshops:

IEEE/ACM Workshop on Automated Formal Methods (AFM) 2007, 2008;

ECOOP International Workshop on Aliasing, Confinement and Ownership in object-oriented programming (IWACO) 2003, 2008;

ACM Workshop on Program Analysis and Security (PLAS) 2008;

NIST Static Analysis Summit (SAS II) 2008;

Workshop on Specification and Verification of Component-Based Systems, 2007;

ACM/CCS Workshop on Formal Methods in Security Engineering (FMSE) 2007, 2008;

LICS/ICALP Workshop on Foundations of Computer Security and Automated Reasoning for Security Protocol Analysis (FCS-ARSPA) 2007;

ESEC/FSE Workshop on Specification and Verification of Component-Based Systems (SAVCBS)

²<http://vstte.ethz.ch/index.html> and <http://qpq.csl.sri.com/vsr/>, <http://www.dagstuhl.de/06281/>

2007;
ECOOP Workshop on Program Analysis for Security and Safety (PASSWORD) 2006;
ECOOP Workshop on Formal Techniques for Java-like Languages 2003, 2004, 2005;
UN International Institute for Software Technology Workshop on Formal Aspects of Component Software 2005;
Brazilian Workshop on Formal Methods 2003.

PhD students

My PhD Students: Qi Sun (defended October 2007): *Constraint-based Secure Information Flow Inference for Object-oriented Programs*.

Stan Rosenberg (expected 2009), Chunyu Tang (expected 2011), Andrey Chudnov (expected 2011).

Opponent and PhD examiner for Leonid Mikhajlov, *Software reuse mechanisms and techniques: safety versus flexibility*, Turku University, Finland, supervisor Ralph Back, 1999.

PhD examiner for Hongseok Yang, *Local reasoning for stateful programs*, University Illinois, supervisor Uday Reddy, 2001 (student was supported by my award INT-9813854).

PhD examiner for Noah Torp-Smith, *Advances in Separation Logic*, IT University of Copenhagen, supervisor Lars Birkedal, 2005.

PhD examiner for Yannis Kassios *A theory of object-oriented refinement*, University of Toronto, supervisor Eric Hehner, 2006.

PhD examiner for Mariela Pavlova *Framework for formal verification of Java bytecode against functional and security policies*, University of Nice, supervisor Gilles Barthe, 2007.

Opponent for Daniel Hedin *Program analysis issues in language based security*, Chalmers University of Technology, Gothenberg, supervisor David Sands, 2008.

PhD committee member: *At Stevens:* Mark Wolfskel (1999), Tendü Yogurtcu (2000), Ricardo Medel (2007), Uma Batchu (2007), Jared Cordasco (in progress), and Ching-Ching Techaubol (in progress). *Elsewhere:* Rohit Gheyi, Federal University of Pernambuco, Brazil (2007).

Undergraduate honors theses supervised

Lam Nguyen, Elizabeth Taylor, and Dustin Long.

Associates

Thesis Supervisors: Edsger W. Dijkstra (U. Texas, Austin; deceased) and C. A. R. Hoare (Oxford University, now Microsoft, Cambridge, UK)

Recent Collaborators: Anindya Banerjee (Kansas State University), Mike Barnett and Wolfram Schulte (Microsoft Research, Redmond), Gilles Barthe and Tamara Rezk (INRIA, Sophia-Antipolis), Paulo Borba and Augusto Sampaio (Federal University of Pernambuco,

Brazil), Ana Cavalcanti (University of Kent), John Hughes, Andrei Sabelfeld, and Alejandro Russo (Chalmers University of Technology), Gary T. Leavens and Steve Shaner (Iowa State University), Marco Pistoia (IBM Research), Uday Reddy (University Birmingham), Philip Wadler (University of Edinburgh), Susanne Wetzels (Stevens)

Selected external funding

Towards a Practical Calculus of Object-Oriented Programming. NSF-CNPq Collaborative Research Opportunities INT-9813854: \$200,000 (1999–2001). Sole PI, with subcontractor Uday Reddy, University Illinois (plus collaborative award to Paulo Borba, Augusto Sampaio, and Ana Cavalcanti, Federal University of Pernambuco, Brazil).

CodeBlue senior design team.Telcordia Technologies: approx. \$14,000 (2001). Additional \$15,000 granted 2003 for ongoing research on this case study.

Stanley Fellowship for PhD student Gerald Thompson: \$15,000 stipend plus tuition (2001–2002).

Integrating Confinement and Access Control for Encapsulation. NSF Trusted Computing CCR-0208984: \$168,727 (2002–2004). Sole PI (plus collaborative award to Anindya Banerjee at Kansas State).

Formal Methods for Behavioral Subclassing and Callbacks. NSF Computing Processes and Artifacts CCF-0429894: \$161,995 (2004–2006). Sole PI (plus collaborative award to Gary Leavens at Iowa State).

Stanley Fellowship for PhD student Stan Rosenberg: \$15,000 stipend plus tuition 2005-06, renewed 2006-07.

Collaborative Research: Access Control and Downgrading in Information Flow Assurance. NSF CyberTrust CNS-0627338: \$206,000 (2006–2009). Sole PI (plus collaborative award to Anindya Banerjee at Kansas State).

Collaborative Research: A JML Community Infrastructure –Revitalizing Tools and Documentation to Aid Formal Methods Research. NSF Computing Research Infrastructure CNS-0708330: \$100,000 (2007–2010). Sole PI (plus collaborative awards to five other universities).

Selected talks since 2002

* = invited

I omit my talks in my department’s seminar series and in the Lab for Secure Systems weekly series.

1. * **Lucent Bell Laboratories**, 9 Jan 02. Representation independence, confinement and access control.
2. **29th ACM Symposium on Principles of Programming Languages**, Portland, OR, 17 Jan 02. Representation independence, confinement and access control.

3. * **Java Verification Workshop, Oregon Graduate Center**, Portland, 12 Jan 02. Reasoning about modules: data refinement and simulation.
4. **Stevens Symposium on CyberSecurity and Trusted Software**, 15 Mar 02. Java, access control, and static analysis.
5. **6th Brazilian Symposium on Programming Languages**, Rio de Janeiro, 5 June 02. On a specification-oriented model for object-orientation.
6. * **6th Brazilian Symposium on Programming Languages**, Rio de Janeiro, 6 June 02. Representation independence, confinement and access control.
7. **15th IEEE Computer Security Foundations**, Cape Breton, Nova Scotia, 26 June 02. Secure information flow and pointer confinement in a Java-like language,
8. * **PointerFest, Queen Mary University of London**, 15 Aug 02. Representation independence, confinement and access control.
9. * **Dagstuhl Seminar on Reasoning about Shape**, Dagstuhl, Germany, 5 March 03. Pointer confinement and abstraction.
10. * **Microsoft Research, Redmond**, 12 March 03. Using access control for secure information flow.
11. * **Cornell University, Information Assurance Institute**, 20 March 03. Using access control for secure information flow in a Java-like language.
12. * **Princeton University, CS Seminar**, 12 May 03. Using access control for secure information flow in a Java-like language
13. * **SRI International** (Stanford Research Institute), Menlo Park, CA, 26 June 03. Using access control for secure information flow.
14. **ECOOP Workshop on Formal Techniques for Java-like Programs**, 21 July 03. Ownership: transfer, sharing, and encapsulation.
15. * **Cornell University, CS Seminar**, 29 Aug 03. Using access control for secure information flow—recent advances.
16. **Dagstuhl Seminar on Language Based Security**, 10 Oct 03. Summary talk outlining research agenda that emerged during this week-long event of which I was a co-organizer.
17. * **Microsoft Research, Cambridge, UK**, 19 Nov 03. Object invariants and owner transfer: issues and approaches.
18. * **INRIA Sophia-Antipolis, France**, 26 Nov 03. Object invariants and owner transfer: issues and approaches.
19. * **Microsoft Research, Redmond, WA**, 10 Dec 03. Object invariants and owner transfer: issues and approaches.
20. **Stevens Cybersecurity Symposium**, 26 March 04. After the buffer overflows are all patched: pre-deployment verification of behavioral contracts.
21. **19th IEEE Symposium on Logic in Computer Science**, Turku, Finland, 16 July 04. Towards imperative modules: Reasoning about invariants and sharing of mutable state.

22. **New Jersey Programming Languages Seminar**, 1 Oct 04. Towards imperative modules: reasoning about invariants and sharing of mutable state.
23. * **Lucent Bell Laboratories**, Jan 05 (lost exact date; hosted by K. Namjoshi). Towards imperative modules.
24. * **European Joint Conferences on Theory and Practice of Software: Grand Challenge Workshop on Software Verification, Edinburgh**, 3 April 05. Using auxiliary state to express modular structure.
25. **European Joint Conferences on Theory and Practice of Software: Conference on Fundamental Aspects of Software Engineering**, 7 April 05. Observational purity and encapsulation.
26. **Department undergrad talk**, 13 April 05. Open but secure: program verification and code based access control.
27. **Microsoft Research, Redmond**, 9 June 05. Observational purity.
28. **19th European Conference on Object-Oriented Programming**, Glasgow, Scotland, 29 July 05. State based ownership, reentrance, and encapsulation.
29. **18th International Conference on Theorem Proving in Higher Order Logics**, Oxford, UK, 25 Aug 05. Verifying a secure information flow analyzer.
30. * **IT University of Copenhagen**, 6 Oct 05. State based ownership, reentrance, and encapsulation.
31. **SRI International**, Menlo Park, CA, 3 April 06. Verified Software Theory Panel Report.
32. **SRI International**, Washington DC, 26 April 06. Verified Software Theory Panel Report.
33. * **IBM Research, Distinguished Speaker**, 27 April 06. Types and contracts for specifying and checking information flow.
34. * **Center for Informatics, Federal University of Pernambuco**, Recife, Brazil, 12 June 07. Pointer Confinement, Encapsulation, and Data Refinement.
35. **Dagstuhl seminar: Challenge of Software Verification**, Dagstuhl, Germany, 13 July 06. Verified Software Theory Panel Report.
36. * **ProSec Security Group, Chalmers University of Technology**, Gothenberg, Sweden, 24 July 06. From coupling relations to mated invariants for checking information flow.
37. **Northeast Verification Seminar, New York University**, 13 Oct 06. Secure information flow by mating invariants.
38. **11th European Symposium on Research in Computer Security**, Hamburg, Germany, 19 Sep 06. From coupling relations and mated invariants for checking secure information flow.
39. * **University of Pennsylvania, Computer Security Seminar**, 26 Oct 06. Specifying and verifying information flow policies using relational Hoare logic.
40. **Stevens/Microsoft Workshop for High School Teachers on Teaching CS**, 23 Mar 07. Using DrJava for introductory programming.
41. **Columbia/IBM/Stevens Security and Privacy Day, Columbia U.**, 1 June 07. Information flow verification and declassification.

42. * **Center for Informatics, Federal University of Pernambuco**, Recife, Brazil, 31 July 07. Assertion-based confinement for encapsulation.
43. * **IBM Research**, Hawthorne NY, 6 Sept 2007. Expressive declassification policies and modular static enforcement.
44. * **Harvard University, programming languages seminar**, 17 Oct 2007. Regional logic for local reasoning about global invariants.
45. * **New England Programming Languages Seminar**, Worcester Polytechnic University, 18 Oct 2007. Expressive declassification policies and modular static enforcement.
46. **Mid-Atlantic Programming Language Seminar (MAPLS) in conjunction with NJ-PLS**, U. Maryland, College Park, 30 Nov 2007. Expressive declassification policies and modular static enforcement.
47. * **Seminar on Types, Logics and Semantics for State**, (3–8 February, 2008), Dagstuhl, Germany, 5 Feb 2008. Regional logic: local reasoning, global invariants.
48. * **Computer Science Seminar**, Stony Brook University, 29 Feb 2008. Information flow with reclassification in object-oriented programs: specification, semantics, and modular verification
49. * **Lehman College Math Seminar**, 27 Feb 2008. Can we make software secure and even leakproof?
50. * **Microsoft Research**, Redmond WA, 14 May 2008. Regional logic: local reasoning, global invariants.
51. **IEEE Symposium on Security and Privacy**, 21 May 2008. Expressive declassification policies and modular static enforcement.
52. * **SRI International, formal methods outreach workshop**, 10 June 2008. Some thoughts on formal methods education.
53. **Automated Formal Methods**, workshop of CAV at Princeton, 14 July 2008. Core JML in PVS.
54. * **Computer Science Seminar, Chalmers University of Technology**, Gothenberg, Sweden, 19 Sept 2008. Expressive declassification policies and modular static enforcement.

Papers in refereed workshop proceedings.

(Some have proceedings published only as technical reports, other proceedings are published by a sponsoring organization like ACM or IEEE.)

1. Simulation and class refinement for Java (with Ana Cavalcanti). In *ECOOP 2000 Workshop on Formal Techniques for Java Programs*, S. Drossopoulou, S. Eisenbach, B. Jacobs, G.T. Leavens, P. Muller, and A. Poetzsch-Heffter, eds. Technical Report 269, Fernuniversitat Hagen, 2000.
2. Class refinement for sequential java (with Ana Cavalcanti). In *ECOOP 2001 Workshop on Formal Techniques for Java Programs*, S. Eisenbach, G.T. Leavens, P. Muller, A. Poetzsch-Heffter, E. Poll, eds.
3. High assurance for interactive applications in ad hoc networks, *Proceedings of First International Workshop on Wireless Security Technologies*, London, April 2003 (with Susanne Wetzel).

4. Ownership: transfer, sharing, and encapsulation (with Anindya Banerjee). In ECOOP 2003 Workshop on Formal Techniques for Java Programs, July 2003.
5. 99.44% pure: Useful Abstractions in Specifications (with Mike Barnett, Wolfram Schulte, and Qi Sun[§]). In ECOOP 2004 Workshop on Formal Techniques for Java-like Programs, June 2004.
6. History-based access control and secure information flow (with Anindya Banerjee). Proceedings of the workshop on *Construction and Analysis of Safe, Secure and Interoperable Smart Cards (CASSIS)*, LNCS 3362 (2004) 27–48.
7. Assertion-based encapsulation, invariants and simulations, July 2005 (invited survey paper), in post-proceedings of *Formal Methods for Components and Objects*, LNCS 3657 (2004) 251–273.
8. Modular Reasoning in Object-oriented Programming (position paper), Post-proceedings, *Verified Software: Theories, Tools, and Experiments (VSTTE)*, Bertrand Meyer and James C. P. Woodcock, eds, (2005).
9. Towards a Logical Account of Declassification (with Anindya Banerjee and Stan Rosenberg[§]). In *ACM SIGPLAN workshop on Programming Languages and Analysis for Security (PLAS)* (2007) 61–66.

Technical reports and magazine articles.

1. A common sense management model (by Raymond T. Yeh, David A. Naumann, Roland Mittermeir, Reinhard A. Schlemmer, William S. Gilmore, George E. Sumrall, and John T. Lebaron). *IEEE Software* 8(6) 23–33 (1991)
2. Predicate transformer semantics of a higher order imperative language with record subtypes. Stevens Institute of Technology CS Report 98-01 (1998) 67 pages.
3. A Weakest Precondition Semantics for an Object-oriented Language of Refinement, extended version (with Ana Cavalcanti). Stevens CS Report 99-03 (1999) 64 pages.
4. Soundness of data refinement for a higher order imperative language. Stevens CS Report 99-05 (1999) 39 pages.
5. Deriving sharp rules of adaptation for Hoare logics. S CS Report 99-06 (1999) 16 pages.
6. An ideal model for pointwise relational programming. Stevens CS Report 20-04 (2000) 17 pages.
7. A simple semantics and static analysis for Java security (with Anindya Banerjee). Stevens CS Report 2001-1, 33 pages.
8. Patterns and lax lambda laws for relational and imperative programming. Stevens CS Report 2001-2, 29 pages.
9. codeBLUE: a Bluetooth interactive dance club system (with Dennis Hromin[§], Michael Chladil[§], Natalie Vanatta[§], Farooq Anjum, and Ravi Jain). Stevens CS Report 2002-1, 19 pages.
10. Ownership transfer and abstraction (with Anindya Banerjee). Kansas State University CIS Report 2004-1, 24 pages.
11. Constraint-based secure information flow inference for a Java-like language (with Qi Sun[§] and Anindya Banerjee). Kansas State University CIS Report 2004-2, 15 pages.

12. Machine-checked correctness of a secure information flow analyzer (preliminary report). Stevens CS Report 2004-10, 80 pages.
Distributed with the PVS code defining the programming language, security policy annotations, security analysis implementation, and correctness proof of the latter.
13. State based encapsulation and generics (with Anindya Banerjee). Stevens CS Report 2004-11 (also appears as Kansas State University CIS-TR-2004-5), 41 pages.
14. Behavioral Subtyping, Specification Inheritance, and Modular Reasoning (with Gary Leavens). Iowa State University Report TR-06-20 (2006) 17 pages.
15. Preliminary Definition of Core JML, (with Gary T. Leavens and Stan Rosenberg[§]). Stevens CS Report 2006-07, 21 pages.
Distributed with the PVS code defining the programming language, its type system and semantics, and the core JML specification language. The PVS code also includes the definitions and theorems for the next item.
16. Behavioral Subtyping is Equivalent to Modular Reasoning for Object-oriented Programs (with Gary T. Leavens) Iowa State University Report TR-06-36 (2006) 19 pages.
17. An admissible second order frame rule in region logic, Stevens CS Report 2008-02.