

# STEPHEN B. MAGILL

Department of Computer Science  
Wean Hall, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213  
(412) 512-6498  
smagill@cs.cmu.edu

## OBJECTIVE

---

I am currently finishing a Ph.D. in Computer Science, specializing in program verification, with an emphasis on shape analysis, separation logic, and numeric abstractions for proving safety and termination. Upon graduation, I intend to continue doing research at either an academic, government, or corporate institution.

## EDUCATION

---

Ph.D., Computer Science Carnegie Mellon University	August 2002-Sept. 2009 (expected graduation)	
M.S. Computer Science Carnegie Mellon University	December 2006	
B.S. Computer Science, Applied Mathematics NSTISSI 4011 INFOSEC Professional Certification University of Tulsa, Tulsa OK	May 2002 May 2002	GPA: 3.97/4.00

## COURSEWORK

---

**Computer Science:** Proving Program Termination, Semantics of Concurrency, Type Systems, Refinement Types, Specification and Verification, Model Checking, Separation Logic, Automated Theorem Proving, Constructive Logic, Complexity Theory

**Mathematics:** Mathematical Logic, Calculus III, Discrete Mathematics, Differential Equations, Linear Algebra, Real Analysis, Modern Algebra, Differential Geometry, Topology, Statistics

## UNIVERSITY RESEARCH EXPERIENCE

---

**Numeric Abstractions:** My thesis work has built on my earlier research into automated shape analysis with separation logic. I have shown how a modification of the shape analysis procedure can be used to produce numeric abstractions of programs. These abstractions are useful for termination reasoning, computing space and time bounds, and checking memory safety of programs that rely on arithmetic properties of data structures. The resulting algorithms have been implemented in a tool called THOR, which performs static analysis of C programs that manipulate heap-based data structures. An example use of THOR would be to reduce termination of a program traversing a skiplist to termination of a certain (automatically generated) numeric program, which can then be checked with a termination tool. Thanks to the reduction, the termination tool need only support simple imperative programs over integer variables.

**Automating Separation Logic:** My early graduate work focused on using separation logic to perform automated shape analysis. I developed a shape analysis algorithm for doubly-linked lists that uses a restricted fragment of separation logic to represent abstract states. I then extended the procedure with support for programmer-specified inductive data structures, enabling automated proofs of memory safety for programs that use various types of trees, skiplists, etc.

**Cryptographic Protocol Verification:** My undergraduate research focused on cryptographic protocol verification. Together with two graduate students, I developed a process calculus that supports the formal, yet natural verification of protocols. The process calculus was implemented in the Isabelle theorem proving environment.

**Secure Electronic Commerce:** Also as an undergrad, I helped design a digital economy simulation for a course in electronic commerce.

## RELATED WORK EXPERIENCE

---

**MSR Cambridge Intern** (spring 2009): I again interned with Josh Berdine and Byron Cook at Microsoft Research. We worked on a variety of extensions to my thesis work, which involves the automatic generation of numeric abstractions for heap-manipulating programs. For example, the numeric abstractions can be used to generate bounds on memory usage and execution time.

**PGSS Faculty** (summer 2008): I developed and led a summer research project for the Pennsylvania Governor's School for the Sciences, which is a program for gifted high school seniors that takes place every summer at CMU. The project was on computer-assisted improvisation and involved the use of Markov chains to generate music in a given style and a user interface consisting of a Guitar Hero™ controller.

**MSR Cambridge Intern** (summer 2006): I worked with Josh Berdine and Byron Cook at Microsoft Research Cambridge on automatically reasoning about programs that use a combination of heap manipulation and arithmetic.

**Teaching Assistant** (spring 2007): I was a TA for "15-213: Introduction to Computer Systems," taught by Frank Pfenning. This is a sophomore-level required course in the CMU computer science curriculum. My duties included grading, leading a one-hour weekly review session / supplemental lecture, and helping students with their lab work. I also gave one lecture during class while the instructor was out of town.

**Teaching Assistant** (spring 2004): I was a TA for Bob Harper's "Constructive Logic" course. This is a junior-level elective in the CMU computer science curriculum. My duties included writing homework assignments and test questions, grading, and leading a one-hour weekly review session / supplemental lecture.

**Institute for Defense Analyses Research Intern** (summer 2002): I spent the summer at the Center for Computing Sciences SCAMP, a three-month long focused research effort involving NSA and IDA scientists.

**National Security Agency Research Intern** (summer 2001): The program began with a two-week Agency lecture series on modern cryptologic methods and was followed by ten weeks of intensive research with NSA mathematicians. I worked with four other students on the analysis of a cryptographic algorithm. Personal Security Clearance: Top Secret/Special Intelligence.

**University of Tulsa Research Assistant** (summers 1999, 2000): During the summers, I continued my undergraduate DoD-sponsored protocol verification research full-time as a paid research assistant.

## PUBLICATIONS

---

Stephen Magill, Ming-Hsien Tsai, Peter Lee, and Yih-Kuen Tsay, "Automatic Numeric Abstractions for Heap-Manipulating Programs," submitted to POPL 2010.

Byron Cook, Stephen Magill, Mohammad Raza, Jiri Simsa, Satnam Singh, "Making Fast Hardware with Separation Logic," submitted to POPL 2010.

Byron Cook, Ashutosh Gupta, Stephen Magill, Andrey Rybalchenko, Jiří Šimša, Satnam Singh, and Viktor Vafeiadis. “Finding heap-bounds for hardware synthesis,” FMCAD 2009.

Stephen Magill, Ming-Hsien Tsai, Peter Lee, and Yih-Kuen Tsay, “THOR: A Tool for Reasoning about Shape and Arithmetic,” 20<sup>th</sup> International Conference on Computer Aided Verification (CAV), 2008.

Stephen Magill, Josh Berdine, Edmund Clarke, and Byron Cook, “Arithmetic Strengthening for Shape Analysis,” 14th International Static Analysis Symposium (SAS), 2007.

Stephen Magill, Aleksandar Nanevski, Edmund Clarke, Peter Lee, “Inferring Invariants in Separation Logic for Imperative List-processing Programs,” Third Workshop on Semantics, Program Analysis, and Computing Environments for Memory Management (SPACE), 2006.

Kevin Donnelly, Tyler Gibson, Neel Krishnaswami, Stephen Magill, Sungwoo Park, “The Inverse Method for the Logic of Bunched Implications,” Proceedings of the Eleventh International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2004), pp. 466-480, 2005.

S. Magill, B. Skaggs, M. Papa and J. Hale, “Implementation and verification of programmable security,” Proceedings of the Sixteenth International Conference on Data and Applications Security (DBSec 2002), Kings College, Cambridge, UK, pp. 285-299, 2002.

B. Skaggs, S. Magill, M. Papa, J. Hale and S. Sheno, “Security service extensions for Java,” Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, pp. 46-53, 2001.

M. Papa, O. Bremer, S. Magill, J. Hale and S. Sheno, “Simulation and analysis of cryptographic protocols,” in Data and Applications Security: Developments and Directions, B. Thuraisingham, R. van de Riet, K. Dittrich and Z. Tari (Eds.), Kluwer, Boston, MA, pp. 89-100, 2001.

S. Magill, B. Enochs, D. Schulte, C. Campbell, G. Manes and J. Hale, “A digital economy simulator,” Proceedings of the Fifth International Conference on Computer Science and Informatics (JCIS 2000), Atlantic City, NJ, pp. 474-475, 2000.

## HONORS AND AWARDS

---

### National Honors

National Science Foundation Graduate Fellowship, 2002: One of 96 to receive this award for work in mathematics and computer science.

National Defense Science and Engineering Graduate Fellowship, 2002: One of 193 national recipients of this award. Provides tuition and stipend for graduate work in science or engineering.

Barry M. Goldwater Scholarship, 2000: One of approximately 300 science and engineering majors nationwide to receive this award for academic and research excellence. Of these, only 15 received the award for work in computer science.

Participant, NSA Director’s Summer Program in Cryptologic Mathematics, 2001: One of 25 advanced mathematics students selected for this prestigious summer program.

### University Honors

Mortar Board Honor Society (Community Service Chair and Honored Initiate), Tau Beta Pi Honor

Society (Secretary), Phi Kappa Phi Honor Society, President's Honor Roll, Top 20 Freshman Award  
TU Presidential Scholarship, College of Engineering Scholarship, Provost's Scholarship, Oklahoma  
Regent's Scholarship

## REFERENCES

---

**Dr. Peter Lee**, Head of Computer Science Department and Professor of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213; petel@cs.cmu.edu, Tel (412) 268-3049

**Dr. John C. Reynolds**, Professor of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213; jcr@cs.cmu.edu, Tel (412) 268-3057

**Dr. Stephen Brookes**, Professor of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213; brookes@cs.cmu.edu, Tel (412) 268-8820

**Dr. Byron Cook**, Researcher, Microsoft Research Cambridge Lab, 7 J J Thomson Ave, Cambridge CB3 0FB, UK; bycook@microsoft.com, Tel +44 1223 479700

**Dr. Josh Berdine**, Researcher, Microsoft Research Cambridge Lab, 7 J J Thomson Ave, Cambridge CB3 0FB, UK; jjb@microsoft.com, Tel +44 1223 479700

**Dr. Sujeet Sheno**i, Oliphant Professor of Computer Science, University of Tulsa, 600 S. College Avenue, Tulsa, OK 74104; sujeet@utulsa.edu, Tel (918) 631-2628