Olin Shivers

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

617 253-1428
617 253-3249 (Fax)
`shivers@ai.mit.edu`
`http://www.ai.mit.edu/~shivers`

Research Interests

My principal research interests lie in the interaction between systems and programming languages, primarily higher-order typed languages.

Education

May 1991 Carnegie Mellon University

Awarded Ph.D. in Computer Science.
Thesis title: *Control-Flow Analysis of Higher-Order Languages.*
Advisors: Professors Peter Lee and Allen Newell.
1987–1991 Thesis work. Research advisor: Professor Peter Lee.
1983–1987 Member of Soar project. Research advisor: Professor Allen Newell.

May 1983 Yale University

Awarded B.S. *cum laude* with double major in Computer Science and Mathematics. Strong background in EE and Physics.

Thesis Overview

I develop techniques for analysing the control-flow structure of languages with first-class procedures and side-effects, such as Scheme or ML. I use the method of non-standard abstract semantic interpretations on a CPS intermediate representation. In the dissertation I (1) develop the control-flow analysis, (2) prove formal properties of the analysis, (3) develop several program optimisations based on the analysis (*e.g.*, induction variable elimination, type recovery), and (4) demonstrate a working implementation of the analysis and optimisation algorithms.

Professional Experience

Spring 1996– MIT AI Lab

Research Scientist, Express project.

Project leader. Supervise the group's doctoral and master's students. Principal author and PI or co-PI of the two DARPA contracts and one NSF grant that fund the project. Express is concerned with the interaction between operating systems, advanced programming languages, compilers, and formal semantics. Our principal research vehicle is ML/OS, an implementation of SML that runs on a bare processor with no OS support.

Fall 1993–Winter 1995 MIT LCS

Research Scientist, Personal Information Architecture group.

1992–Summer 1993 University of Hong Kong

Member of the faculty, Computer Science Department.

Fall 1991 AT&T Bell Labs

Post-doc in ML-related research with David MacQueen's Software Principles group.

Summer 1986 ITC, CMU

Ported Orbit Scheme compiler to the IBM RT PC.

Summer 1985 *Centre Mondial Informatique et Ressource Humaine*, Paris

Designed and implemented object-oriented 3D-graphics system.

Summer 1984 DEC Western Research Lab

Member of Jonathan Rees's T group. The group designed and implemented version 3 of the Scheme dialect T, and its optimising compiler, ORBIT. Worked on flow analysis, language design, runtime internals, and the linker.

Summer 1982 MIT AI Lab

Worked for Prof. Randy Davis' Hardware Troubleshooting project. Wrote constraint compiler for DPL implementation language. Designed language extensions. Systems/language work on the Lisp programming environment.

Teaching Experience

1997 University of Århus, Denmark

Invited mini-course for graduate students on systems programming in advanced programming languages.

1996 MIT

Taught 6.821, graduate student course on programming language semantics, with Prof. Gifford. Received excellent student evaluations.

1992,1993 University of Hong Kong

Classes included an "Advanced Language Implementation Technology" graduate seminar, "Principles of Programming Languages" undergraduate class, and introductory programming in C. Class sizes varied from 25 to 400. Received excellent student evaluations.

Spring 1986 Carnegie Mellon University

Taught *Introduction to Lisp* undergraduate course. Had sole responsibility for course: wrote and gave all lectures, wrote and graded all assignments and programming projects, assigned final grades. Received outstanding student evaluations.

Publications

The Flux OSKit: A substrate for kernel and language research. Bryan Ford, Godmar Back, Greg Benson, Jay Lepreau, Albert Lin and Olin Shivers. Proceedings of the Sixteenth ACM Symposium on Operating Systems Principles (SOSP-16), October 1997, Saint-Malo, France.

Automatic management of operating system resources. In *Proceedings of the Second ACM SIGPLAN International Conference on Functional Programming (ICFP '97)*, June 1997, Amsterdam.

Continuations and threads: Expressing machine concurrency directly in advanced languages. In *Proceedings of the Second ACM SIGPLAN Workshop on Continuations,* January 1997, Paris. Also available as BRICS Notes Series NS-96-13, University of Århus, Denmark.

A universal scripting framework, or Lambda: the ultimate "little language." In *Concurrency and Parallelism, Programming, Networking, and Security,* Lecture Notes in Computer Science #1179, pages 254–265, Editors Joxan Jaffar and Roland H. C. Yap, 1996, Springer.

Supporting dynamic languages on the Java virtual machine. In *Proceedings of the Dynamic Objects Workshop*, May 1996, Boston, Also available as technical report AIM-1576, MIT Artificial Intelligence Lab.

A Scheme Shell. *Lisp and Symbolic Computation*, to appear. (Also available as Technical report TR-94-10, Dept of Computer Science, University of Hong Kong, and Technical report TR-635, Laboratory for Computer Science, MIT.)

The semantics of Scheme control-flow analysis. In *Proceedings of the First ACM SIGPLAN and IFIP Symposium on Partial Evaluation and Semantics-Based Program Manipulation*, published as *SIGPLAN Notices* 26(9):190–198, Association for Computing Machinery, September 1991. (Also available as Technical Report CMU-CS-91-119, CMU School of Computer Science, Pittsburgh, Penn.)

Control-Flow Analysis of Higher-Order Languages. Ph.D. dissertation, Carnegie Mellon University, May 1991. Technical Report CMU-CS-91-145, School of Computer Science.

CPS data-flow analysis example. Scheme Flow Analysis Working note #1 (May 1, 1990).

Useless-variable elimination. Scheme Flow Analysis Working note #2 (April 27, 1990). Proceedings of the Workshop on Static Analysis of Equational, Functional and Logic Programs, Bordeaux, France, October 1991.

Super-β: Copy, constant, and lambda propagation in Scheme. Scheme Flow Analysis Working note #3 (May 1, 1990).

Data-flow analysis and type recovery in Scheme. Technical Report CMU-CS-90-115, CMU School of Computer Science, Pittsburgh, Penn., March 1990. (Also appears as chapter 3 of *Topics in Advanced Language Implementation*, ed. Peter Lee, MIT Press, 1991.)

Constraint Satisfaction and Problem Space Structure in Soar, In *Proceedings of the Fifth Soar Workshop,* September 17, 1988.

Control-flow analysis in Scheme. In *Proceedings of the SIGPLAN '88 Conference on Programming Language Design and Implementation*, June 1988. (Also available as Technical Report ERGO-88-60, CMU School of Computer Science, Pittsburgh, Penn.)

Varieties of learning in Soar: 1987, by David Steier, *et al.* In *Proceedings of the Fourth International Workshop on Machine Learning*, Pat Langley (editor), Morgan Kaufmann, June 1987.

Modelling Cryptarithmetic Puzzle Solving for Subject S3 of *Human Problem Solving*, In *Proceedings of the Soar Spring 87 Workshop*, June 20, 1987.

Constraint Propagation and Macro-compilation in Soar, In *Proceedings of the Soar Fall 86 Workshop*, November 22, 1986.

Other

Invited talks (partial list): Indiana University, NEC, Yale, Academia Sinica, Peking Software Institute, INRIA, University of Århus, IFIP WG2.8, PEPM '99.

Member ICFP steering committee.

Member FPCA '95 program committee.

Personal Data

Born September 1961 in Atlanta, Ga.
Single.
Health excellent.
Conversational French and Mandarin; some written Chinese.

References

Dr. Thomas Knight
Prof. Gerald Sussman
MIT AI Lab
545 Technology Square
Cambridge, Mass. 02139
tk@ai.mit.edu
gjs@ai.mit.edu

Professor Peter Lee
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
412-268-2565
petel@cs.cmu.edu

Professor Olivier Danvy
University of Århus
Department of Computer Science
Ny Munkegade
DK-8000 Århus C, Denmark
danvy@brics.dk

Professor Francis Y. L. Chin
Computer Science Department
University of Hong Kong
Pokfulam Road
Hong Kong
chin@cs.hku.hk